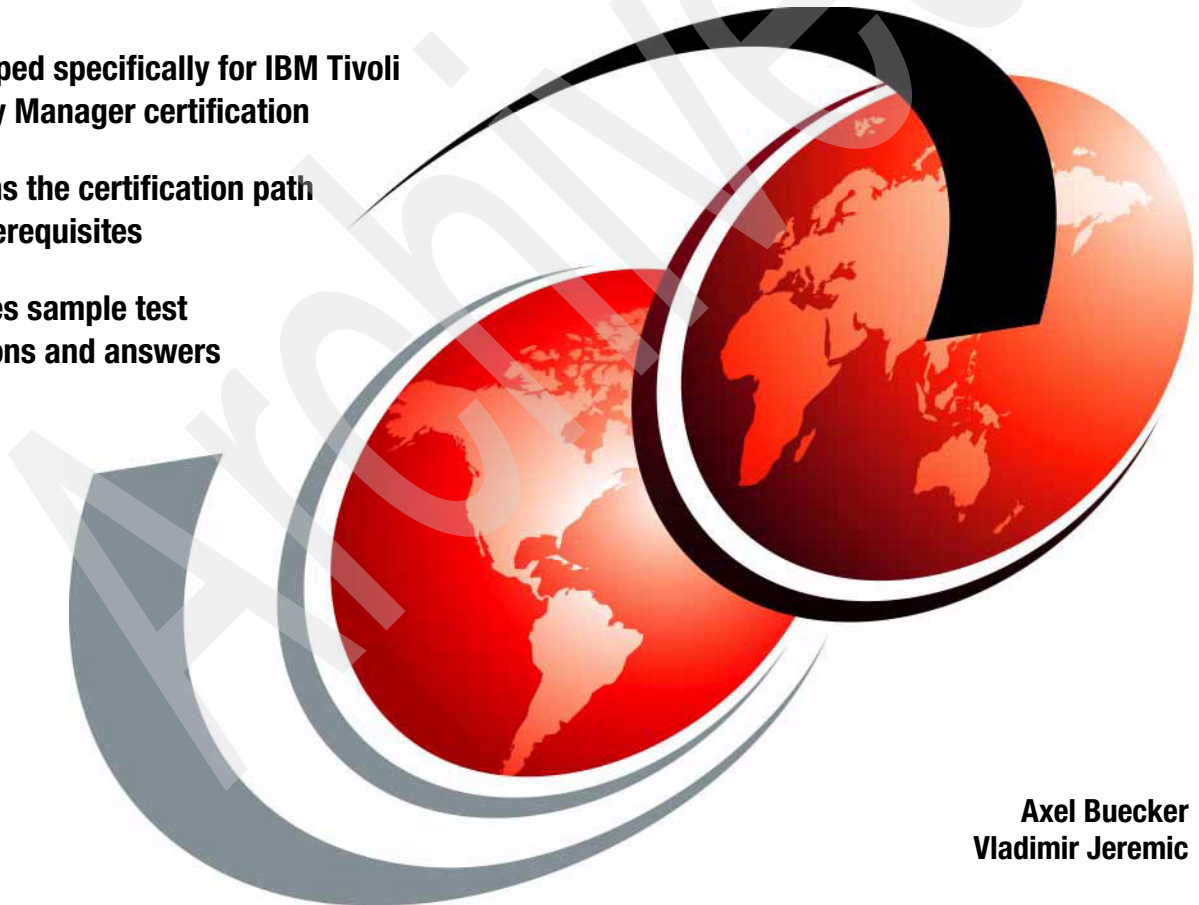


Certification Study Guide: IBM Tivoli Identity Manager Version 5.0

Developed specifically for IBM Tivoli
Identity Manager certification

Explains the certification path
and prerequisites

Includes sample test
questions and answers



Axel Buecker
Vladimir Jeremic



International Technical Support Organization

**Certification Study Guide: IBM Tivoli Identity
Manager Version 5.0**

February 2009

Archived

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

Second Edition (February 2009)

This edition applies to Version 5 of IBM Tivoli Identity Manager.

© Copyright International Business Machines Corporation 2005, 2009. 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

Notices	ix
Trademarks	x
Preface	xi
The team that wrote this book	xi
Become a published author	xii
Comments welcome	xii
Chapter 1. Certification overview	1
1.1 IBM Professional Certification Program	2
1.1.1 Benefits of certification	3
1.1.2 IBM Tivoli Software Professional Certification	4
1.2 IBM Tivoli Identity Manager V5.0 certification	7
1.2.1 Job description and target audience	7
1.2.2 Key areas of competency	7
1.2.3 Required prerequisites	8
1.2.4 Test 934 objectives	9
1.3 Recommended educational resources	27
1.3.1 Courses	27
1.3.2 Publications	34
Chapter 2. Planning	39
2.1 Overview	40
2.2 Organization structure design	41
2.3 Service design	43
2.4 Entities design	43
2.4.1 Users, accounts, and attributes	44
2.4.2 Passwords	45
2.4.3 Group membership	46
2.4.4 Managed systems and applications	46
2.5 Life cycle management design	47
2.5.1 The registration/creation cycle	48
2.5.2 The provisioning cycle	48
2.5.3 The modification cycle	48
2.5.4 The termination cycle	49
2.5.5 Life cycle management	49
2.6 E-mail management design	50
2.6.1 Notification templates	50
2.6.2 Post office	51

2.7 IBM Tivoli Identity Manager group design	51
2.8 Provisioning policies design	53
2.9 Workflow design	55
2.10 Identity policy design	57
2.11 Password policies design	57
2.12 Security model design	59
2.12.1 Access provisioning models	60
2.12.2 Role-based access control	61
2.13 Customization design	62
2.13.1 Graphical user interface	62
2.14 System architecture	69
2.14.1 High availability	70
2.14.2 Archival and backup	71
2.15 Adapter project plan	72
2.16 IBM Tivoli Identity Manager project planning	73
Chapter 3. Installation	79
3.1 IBM Tivoli Identity Manager components overview	80
3.2 SSL communication overview	83
3.2.1 Certificate and key formats	85
3.2.2 SSL handshake	86
3.3 Installation process	88
3.4 Adapter installation and configuration	93
3.4.1 RMI-based adapters	95
3.4.2 ADK-based adapters	97
Chapter 4. Implementation	101
4.1 IBM Tivoli Identity Manager components overview	102
4.2 Organization tree	103
4.2.1 Organization tree elements	104
4.2.2 Organizational roles	104
4.2.3 Placement rules	105
4.3 Tivoli Identity Manager user types	106
4.4 Services	107
4.4.1 Identity feed service types	108
4.4.2 Account service types	109
4.4.3 Reconciliation	110
4.5 Policy	113
4.5.1 Identity policy	113
4.5.2 Password policy	116
4.5.3 Provisioning policy	119
4.5.4 Service selection policy	125
4.5.5 Adoption policy	126

4.5.6	Recertification policy	126
4.5.7	Account defaults	127
4.6	Workflows	128
4.6.1	Account request workflow	128
4.6.2	Access request workflow	129
4.6.3	Operation workflow	129
4.6.4	Workflow elements	131
4.6.5	Workflow notification properties	136
4.7	Tivoli Identity Manager groups	136
4.8	Access control item	138
4.8.1	Conflicts between multiple ACIs and Tivoli Identity Manager groups	141
4.9	Views	141
4.10	Auditing	142
4.11	Reporting	143
4.12	Post office	147
4.12.1	E-mail notifications templates	150
4.13	Configuring commonly used system properties	151
4.14	Modifying system properties manually	153
4.15	Modifying system properties with the GUI	159
4.16	User interface customization	160
4.16.1	Administrative console customization	161
4.16.2	Self-service user interface customization	163
4.17	Directory server	165
Chapter 5.	Data management	171
5.1	Identity feed overview	172
5.2	Initial identity feed preparation	172
5.3	Types of initial identity feed	173
5.3.1	Manual identity feed	174
5.3.2	Comma-Separated Value (CSV) Identity Feed Service	174
5.3.3	DSML Identity Feed Service	174
5.3.4	Windows Server Active Directory Identity Feed Service	175
5.3.5	INetOrgPerson Identity Feed	175
5.3.6	IDI Data Feed Service	175
5.3.7	Programming approach	176
5.3.8	Self-registration	176
5.3.9	Placement rule	176
5.3.10	Attribute mapping file	176
5.3.11	Enabling workflow for identity feeds	177
5.4	Deploying initial identity feed with existing accounts	177
Chapter 6.	Troubleshooting	179
6.1	Troubleshooting problems	180

6.1.1	Troubleshooting installation errors	180
6.1.2	Troubleshooting operation errors	181
6.2	Log files	182
6.2.1	Types of logs	182
6.2.2	Tivoli Identity Manager Server operation log files	186
6.3	Traces	189
6.3.1	Server tracing	189
6.3.2	Applet tracing	191
6.4	Adapter troubleshooting	191
6.5	Diagnostic tools	192
6.5.1	Diagnosing completed requests with the audit log	192
6.5.2	Viewing log file data	192
6.5.3	Tivoli Identity Manager serviceability tool	193
6.6	Additional resources	194
Chapter 7. Production		195
7.1	Data migration	196
7.1.1	Export	196
7.1.2	Import	198
7.1.3	Additional considerations	198
7.2	Reconciliation	199
7.2.1	Reconciliation of manual service	202
7.3	Recycle bin periodical maintenance	202
Chapter 8. Maintenance		205
8.1	Performance monitoring and tuning	206
8.1.1	Performance monitoring	206
8.1.2	Tuning	207
8.2	Migration	210
8.2.1	Migration planning and preparation phase	211
8.2.2	Tivoli Identity Manager Server upgrade phase	211
8.2.3	Post-upgrade phase	212
8.3	Fix pack installation	215
Appendix A. Sample questions		217
	Questions	218
	Answer key	221
Appendix B. Definitions of path variables		223
Related publications		227
	IBM Redbooks publications	227
	Other publications	227
	Online resources	228

How to get IBM Redbooks publications	228
Help from IBM	228
Index	229

Archived

Archived

Notices

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

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

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

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

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

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

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

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

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


COPYRIGHT LICENSE:

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

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.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:

AIX 5L™	Lotus®	System x®
AIX®	OS/400®	System z®
DB2 Universal Database™	RACF®	Tivoli®
DB2®	Redbooks®	WebSphere®
Domino®	Redbooks (logo)  ®	z/OS®
IBM®	System p®	

The following terms are trademarks of other companies:

Acrobat, Adobe, and Portable Document Format (PDF) are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

SUSE, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

EJB, Java, JavaMail, JavaScript, JDBC, JRE, JSP, JVM, Solaris, Sun, Sun Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Active Directory, Internet Explorer, Microsoft, SQL Server, Windows NT, Windows Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Itanium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

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

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

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

Preface

This IBM® Redbooks® publication is a study guide for the “IBM Certified Deployment Professional - IBM Tivoli® Identity Manager V5.0” certification test, test number 934, and is meant for those who want to achieve IBM Certifications for this specific product.

The IBM Certified Deployment Professional - IBM Tivoli Identity Manager V5.0 certification, offered through the Professional Certification Program from IBM, is designed to validate the skills required of technical professionals who work in the implementation of the IBM Tivoli Identity Manager Version 5.0 product.

This book provides a combination of theory and practical experience needed for a general understanding of the subject matter. It also provides sample questions that will help in the evaluation of personal progress and provide familiarity with the types of questions that will be encountered in the exam.

This book does not replace practical experience, and it is not designed to be a stand-alone guide for any subject. Instead, it is an effective tool that, when combined with education activities and experience, can be a very useful preparation guide for the exam.

The team that wrote this book

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

Axel Buecker is a Certified Consulting Software IT Specialist at the ITSO, Austin Center. He writes extensively and teaches IBM classes worldwide on areas of software security architecture and network computing technologies. He holds a degree in Computer Science from the University of Bremen, Germany. He has 22 years of experience in a variety of areas related to workstation and systems management, network computing, and e-business solutions. Before joining the ITSO in March 2000, Axel worked for IBM in Germany as a Senior IT Specialist in Software Security Architecture.

Vladimir Jeremic is a Managing Consultant with the IBM Global Services Security and Privacy Practice where he focuses on architecture and implementation of the Tivoli Security portfolio. He has over ten years of experience in the IT field related to security, networking, and programming. He is a Tivoli Certified Professional and holds a BS E.E. degree from the University of

Novi Sad, in Serbia. He has experience in designing and developing learning materials. Vladimir also participated in developing several IBM Redbooks publications related to the IBM Tivoli Security portfolio.

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

The IBM Tivoli Education Development Team
IBM US

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

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 in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an e-mail to:
redbooks@us.ibm.com
- ▶ Mail your comments to:

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

Certification overview

This chapter provides an overview of the skill requirements needed to obtain an IBM Advanced Technical Expert certification. The following sections are designed to provide a comprehensive review of specific topics that are essential for obtaining the certification:

- ▶ IBM Professional Certification Program
- ▶ IBM Tivoli Identity Manager V5.0 certification
- ▶ Recommended educational resources

1.1 IBM Professional Certification Program

Having the right skills for the job is critical in the growing global marketplace. IBM Professional Certification, designed to validate skill and proficiency in the latest IBM solution and product technology, can help provide that competitive edge. The IBM Professional Certification Program Web site is available at:

<http://www.ibm.com/certify/index.shtml>

The Professional Certification Program from IBM offers a business solution for skilled technical professionals seeking to demonstrate their expertise to the world.

The program is designed to validate your skills and demonstrate your proficiency in the latest IBM technology and solutions. In addition, professional certification can help you excel at your job by giving you and your employer confidence that your skills have been tested. You might be able to deliver higher levels of service and technical expertise than non-certified employees and move on a faster career track. Professional certification puts your career in your control.

The certification requirements are tough, but not impossible. Certification is a rigorous process that differentiates you from everyone else. The mission of IBM Professional Certification is to:

- ▶ Provide a reliable, valid, and fair method of assessing skills and knowledge
- ▶ Provide IBM with a method of building and validating the skills of individuals and organizations
- ▶ Develop a loyal community of highly skilled certified professionals who recommend, sell, service, support, and use IBM products and solutions

The Professional Certification Program from IBM has developed certification role names to guide you in your professional development. The certification role names include IBM Certified Specialist, IBM Certified Solutions/Systems Expert, and IBM Certified Advanced Technical Expert for technical professionals who sell, service, and support IBM solutions.

For technical professionals in application development, the certification roles include IBM Certified Developer Associate and IBM Certified Developer. IBM Certified Instructor certifies the professional instructor.

The Professional Certification Program from IBM provides a structured program leading to an internationally recognized qualification. The program is designed for flexibility by enabling you to select your role, prepare for and take tests at your own pace, and, in some cases, select from a choice of elective tests that are best

suited to your abilities and needs. Some roles also offer a shortcut by giving credit for a certification obtained in other industry certification programs.

You might be a network administrator, systems integrator, network integrator, solution architect, solution developer, value-added reseller, technical coordinator, sales representative, or educational trainer. Regardless of your role, you can start charting your course through the Professional Certification Program from IBM today.

1.1.1 Benefits of certification

Certification is a tool to help objectively measure the performance of a professional on a given job at a defined skill level. Therefore, it is beneficial for individuals who want to validate their own skills and performance levels, their employees, or both. For optimum benefit, the certification tests must reflect the critical tasks required for a job, the skill levels of each task, and the frequency by which a task needs to be performed. IBM prides itself in designing comprehensive, documented processes that ensure that IBM certification tests remain relevant to the work environment of potential certification candidates.

In addition to assessing job skills and performance levels, professional certification can also provide such benefits as:

- ▶ For employees:
 - Promotes recognition as an IBM Certified Professional
 - Helps to create advantages in interviews
 - Assists in salary increases, corporate advancement, or both
 - Increases self-esteem
 - Provides continuing professional benefits
- ▶ For employers:
 - Measures the effectiveness of training
 - Reduces course redundancy and unnecessary expenses
 - Provides objective benchmarks for validating skills
 - Makes long-range planning easier
 - Helps to manage professional development
 - Aids as a hiring tool
 - Contributes to competitive advantage
 - Increases productivity, morale, and loyalty
- ▶ For Business Partners and consultants:
 - Provides independent validation of technical skills
 - Creates competitive advantage and business opportunities

- Enhances prestige of the team
- Contributes to IBM requirements for various IBM Business Partner programs

Specific benefits might vary by country (region) and role. In general, after you become certified, you should receive the following benefits:

▶ Industry recognition

Certification might accelerate your career potential by validating your professional competency and increasing your ability to provide solid, capable technical support.

▶ Program credentials

As a certified professional, you receive an e-mail with your certificate of completion and the certification mark associated with your role for use in advertisements and business literature. You can also request a hardcopy certificate, which includes a wallet-size certificate.

The Professional Certification Program from IBM acknowledges the individual as a technical professional. The certification mark is for the exclusive use of the certified individual.

▶ Ongoing technical vitality

IBM Certified Professionals are included in mailings from the Professional Certification Program from IBM.

1.1.2 IBM Tivoli Software Professional Certification

The IBM Tivoli Professional Certification Program offers certification testing that sets the standard for qualified product consultants, administrators, architects, and partners.

The program also offers an internationally recognized qualification for technical professionals who are seeking to apply their expertise in today's complex business environment. The program is designed for those who implement, buy, sell, service, and support IBM Tivoli solutions and who want to deliver higher levels of service and technical expertise.

Whether you are an IBM Tivoli customer, partner, or technical professional wanting to put your career on the fast track, you can start your journey to becoming an IBM Tivoli Certified Professional today.

Benefits of being IBM Tivoli certified

Tivoli Certification has the following benefits:

► For the individual:

- IBM Certified certificate and use of logos on business cards

Note: Certificates are sent by e-mail; however, you can also request a paper copy of the certificate and a laminated wallet card by sending an e-mail to <mailto:certify@us.ibm.com>.

- Recognition of your technical skills by your peers and management
- Enhanced career opportunities
- Focus for your professional development

► For the IBM Business Partner:

- Confidence in the skills of your employees
- Enhanced partnership benefits from the IBM Business Partner Program
- Higher rates for billing out your employees
- Stronger customer proposals
- Demonstration of the depth of technical skills available to prospective customers

► For the customer:

- Confidence in the services professionals handling your implementation
- Ease of hiring competent employees to manage your Tivoli environment
- Enhanced return on investment (ROI) through more thorough integration with Tivoli and third-party products
- Ease of selecting an IBM Tivoli Business Partner that meets your specific needs

Certification checklist

Here is the certification checklist:

1. Select the certification that you want to pursue.
2. Determine which tests are required by reading the certification role description.
3. Prepare for the test, using the following resources:
 - Test objectives
 - Recommended educational resources
 - Sample/Assessment test
 - Other reference materials
 - Opportunities for experience

Note: These resources are available from each certification description page and from the Test information page.

4. Register to take a test, by contacting one of our worldwide testing vendors:
 - Thomson Prometric
 - Pearson Virtual University Enterprises (VUE)

Note: When providing your name and address to the testing vendor, be sure to specify your name exactly as you want it to appear on your certificate.

5. Take the test. Be sure to keep the Examination Score Report that is provided when you complete the test as your record of taking the test.

Note: After you take the test, the results and demographic data (such as name, address, e-mail, and phone number) are sent from the testing vendor to IBM for processing (allow two to three days for transmittal and processing). After all the tests that are required for a certification are passed and received by IBM, your certificate will be issued.

6. Repeat steps 3 through 5 until you have completed successfully all the required tests for the certification. If there are additional requirements (such as another vendor certification or exam), follow the instructions on the certification description page to submit these requirements to IBM.
7. After you meet the requirements, you will receive an e-mail asking you to accept the terms of the IBM Certification Agreement.
8. Upon your acceptance, you receive an e-mail with the following deliverables:
 - A Certification Certificate in PDF format, which can be printed in either color or black and white
 - A set of graphic files containing the IBM Professional Certification mark that is associated with the certification achieved
 - Guidelines for the use of the IBM Professional Certification mark
9. To avoid an unnecessary delay in receiving your certificate, ensure that your current e-mail is on file by keeping your profile up to date. If you do not have an e-mail address on file, your certificate will be sent by postal mail.

After you receive a certificate by e-mail, you can also contact IBM at <mailto:certify@us.ibm.com> to request a hardcopy certificate sent by postal mail.

Note: IBM reserves the right to change or delete any portion of the program, including the terms and conditions of the IBM Certification Agreement, at any time without notice. Some certification roles offered through the IBM Professional Certification Program require recertification.

1.2 IBM Tivoli Identity Manager V5.0 certification

In this section, we categorize the certification process for Tivoli Identity Manager (also known as *Identity Manager*).

Important: IBM offers the following promotion code, which is good for a 15% discount on the indicated Tivoli certification exams if taken at any Thomson Prometric testing center:

- ▶ Code: 15T934
- ▶ Percentage off: 15%
- ▶ Valid for exams: 000-934

1.2.1 Job description and target audience

An IBM Certified Deployment Professional is a technical professional who is responsible for planning, installation, configuration, data management, troubleshooting, rollout to production, maintenance, and upgrade of a Tivoli Identity Manager V5.0 solution. This person is expected to perform these tasks without assistance or with only limited assistance from peers, product documentation, and support resources.

1.2.2 Key areas of competency

The following key areas of competency are required to pass the Certification Test 934.

- ▶ Describe the Tivoli Identity Manager V5.0 architecture and components.
- ▶ Implement a Tivoli Identity Manager V5.0 solution based on customer requirements and environment based on a solution design.
- ▶ Install and configure prerequisites to Tivoli Identity Manager V5.0.
- ▶ Install and configure Tivoli Identity Manager V5.0 infrastructure components.

- ▶ Use available interfaces to configure and administer the Tivoli Identity Manager V5.0 environment.
- ▶ Perform performance tuning and problem determination for Tivoli Identity Manager V5.0.

1.2.3 Required prerequisites

The required prerequisites needed to pass the Certification Test 934 include the following abilities.

- ▶ Understanding of custom changes and extensions to Tivoli Identity Manager V5.0-Skill Level 3.¹
- ▶ Understanding of Tivoli Identity Manager V5.0 architecture and components-Skill Level 3.
- ▶ Experience installing and configuring Tivoli Identity Manager V5.0-Skill Level 3.
- ▶ Knowledge of Tivoli Identity Manager V5.0 prerequisite software-Skill Level 3.
- ▶ Understanding of the migration process from Tivoli Identity Manager V4.6 to V5.0-Skill Level 3.
- ▶ Understanding of business processes-Skill Level 3.
- ▶ Knowledge of LDAP and LDAP expressions-Skill Level 3.
- ▶ Knowledge of IBM Tivoli Directory Integrator-Skill Level 3.
- ▶ Knowledge of basic security concepts (encryption using keys, SSL, HTTPS) -Skill Level 2.
- ▶ Perform installation and updates of single server and clustered installation of WebSphere® Application Server-Skill Level 2.
- ▶ Understanding of JavaScript™, XML, DSML-Skill Level 2.
- ▶ Knowledge of operating systems-Skill Level 2.
- ▶ Knowledge of administrative models and user management-Skill Level 2.
- ▶ Knowledge of cascading style sheets-Skill Level 2.

¹ The skill levels are represented as follows:

Level 1 - Basic Skill/Knowledge: Familiarity with basic functionality and concepts. Might need to rely on assistance from documentation or other resources.

Level 2 - Working Skill/Knowledge: Working knowledge of functionality and concepts. Can use product or explain concepts with little or no assistance.

Level 3 - Advanced Skill/Knowledge: Substantial experience with functionality or concepts. Can teach others how to use functionality or explain concepts.

Level 4 - Expert Skill/Knowledge: Extensive and comprehensive experience with functionality or concepts. Can create or customize code, architecture, or processes.

- ▶ Understanding of Java™-Skill Level 2.
- ▶ Understanding of basic system architecture design-Skill Level 2.
- ▶ Perform basic installation and updates of the prerequisite databases (DB2®, Oracle® and SQL Server®) and LDAP directory servers (IBM Tivoli Directory Server and SunOne Server) -Skill Level 1.
- ▶ Knowledge of shell scripting-Skill Level 1.
- ▶ Knowledge of TCP/IP-Skill Level 1.

1.2.4 Test 934 objectives

This test includes the following objective areas:

- ▶ Section 1: Planning
- ▶ Section 2: Installation
- ▶ Section 3: Implementation
- ▶ Section 4: Data management
- ▶ Section 5: Troubleshooting
- ▶ Section 6: Production
- ▶ Section 7: Maintenance
- ▶ Section 8: Enhancements in V5.0

Section 1: Planning

The section provides further information about the planning area of the test:

- ▶ Given the existing organization and reporting structure, gather the requirements and develop the solution so that an organization structure design is created. The emphasis is on being able to perform the following steps:
 - Gather the organization structure requirements
 - Discuss alternatives
 - Formalize the organization structure
 - Document the organization structure
- ▶ Given the desired services list and organization structure design, gather target platforms, business processes and develop the solution so that a Service design is created. The emphasis is on being able to perform the following steps:
 - Gather the services target platforms
 - Define the organization requirements
 - Gather the platform business processes
 - Identify unsupported platforms
 - Document the services requirements

- ▶ Given the existing human resources data and the services design, gather entity requirements and develop the solution so that an entities design is created. The emphasis is on being able to perform the following steps:
 - Validate the Human Resource data
 - Gather the entity requirements
 - Design the entities
 - Document the entity design
- ▶ Given the existing and projected business processes, gather the life cycle management requirements and develop the solution so that a life cycle management design is created. The emphasis is on being able to perform the following steps:
 - Gather the life cycle management requirements
 - Design the life cycle management strategy
 - Document the life cycle design
- ▶ Given the existing and projected business processes, gather the e-mail management requirements and develop the solution so that an e-mail management design is created. The emphasis is on being able to perform the following steps:
 - Determine the e-mail volume and frequency
 - Determine the aggregation policy
 - Determine format and content of the aggregated e-mail
- ▶ Given the existing role information and organization structure design, gather the role requirements and develop the solution so that a roles design is created. The emphasis is on being able to perform the following steps:
 - Gather the role requirements
 - Define the organization requirements
 - Design the high-level role structure
 - Document the role design
- ▶ Given the existing provisioning policies and organization structure design, gather requirements and discuss and formalize the design so that a provisioning policies design is created. The emphasis is on being able to perform the following steps:
 - Gather the policy requirements
 - Define the organization requirements
 - Gather the entitlement requirements
 - Define the membership
 - Design the high-level policy structure
 - Define the service selection policies
 - Document the policy design

- ▶ Given the existing workflows and services design, gather the requirements, including the workflow scope and approach, and develop the solution so that the workflow design is created. The emphasis is on being able to perform the following steps:
 - Gather the workflow requirements
 - Define the workflow scope
 - Design the workflow approach
 - Document the workflow design
- ▶ Given the existing human resources data and the entities, design, analyze, and map the data to the Tivoli Identity Manager LDAP attributes and develop the solution so that the Person/BP Person identity design is created. The emphasis is on being able to perform the following steps:
 - Gather the identity source requirements
 - Analyze the identity source data
 - Map the identity data to Tivoli Identity Manager
 - Document the identity requirements
- ▶ Given the existing identity policies and guidelines, entities design, and identity sources design, gather the identity policy and organizational requirements and develop the solution so that the identity policy design is created. The emphasis is on being able to perform the following steps:
 - Gather the identity policy requirements
 - Define the organizational requirements
 - Design the high-level ID policy approach
 - Document the ID policy design
- ▶ Given the existing password policies and services design, gather the requirements and define the scope so that the password policy design is created. The emphasis is on being able to perform the following steps:
 - Gather the password policy requirements
 - Define the password policy scope
 - Define the password settings
 - Document the password policy design
- ▶ Given the existing application security policies, organization structure design, services design, and entity design, gather the Tivoli Identity Manager access requirements and design groups and ACIs so that the security model design is created. The emphasis is on being able to perform the following steps:
 - Gather the Tivoli Identity Manager access requirements
 - Design the Tivoli Identity Manager groups
 - Design the Tivoli Identity Manager ACIs
 - Document the Tivoli Identity Manager security model

- ▶ Given the proper policies and documentation, gather the customization requirements and determine the feasibility and scope so that the customization design is created. The emphasis is on being able to perform the following steps:
 - Gather the customization requirements
 - Determine the customization feasibility
 - Design the high-level functionality
 - Determine the customization scope
 - Document the server customization design
- ▶ Given the proper documentation, gather the adapter requirements and develop the solution so that a custom adapter design is created. The emphasis is on being able to perform the following steps:
 - Gather the adapter requirements, including account and group access requirements
 - Determine the customization feasibility
 - Design the high-level functionality
 - Determine the customization scope
 - Document the adapter customization design
- ▶ Given the hardware assets list, existing network configuration, and the services design, gather the system architecture requirements and design the solution so that a system architecture document is created. The emphasis is on being able to perform the following steps:
 - Gather the system architecture requirements
 - Design the system architecture
 - Document the system architecture
- ▶ Given the services design and existing project plans, prioritize the platforms and determine the adapter phases so that an adapter project plan is created. The emphasis is on being able to perform the following steps:
 - Prioritize the platforms
 - Group the adapters into phases
 - Determine a timeline for phases
 - Document an adapter rollout plan
- ▶ Given the proper documentation, gather the initial timeline requirements and determine the initial solution rollout timeline so that an initial solution rollout project plan is created. The emphasis is on being able to perform the following steps:
 - Gather the timeline requirements
 - Determine a timeline for production rollout
 - Document the timeline or plan

- ▶ Given the system architecture design and existing backup processes, gather the backup requirements and develop the solution so that a backup and recovery strategy design is created. The emphasis is on being able to perform the following steps:
 - Gather the backup requirements
 - Design a backup strategy
 - Document the backup strategy
- ▶ Given the proper documentation, analyze the current system and upgrade requirements so that an upgrade planning document is created. The emphasis is on being able to perform the following steps:
 - Analyze the current system
 - Determine system changes from an upgrade
 - Analyze the customizations
 - Design the customization upgrade plan
 - Develop an overall upgrade plan
 - Document the upgrade plan
- ▶ Given the proper documentation, analyze the business processes and requirements so that a custom reporting requirements document is created. The emphasis is on being able to perform the following steps:
 - Gather the business requirements
 - Define the reporting data
 - Define the report form
 - Document the requirements
- ▶ Given the existing account recertification process, gather the account recertification requirements and develop the solution so that a recertification design document is created. The emphasis is on being able to perform the following steps:
 - Gather the recertification management requirements
 - Design a recertification management strategy
 - Document the recertification design
- ▶ Given the IT infrastructure definition, the projected user population to be managed, and the business continuity requirements, gather the availability and scalability requirements so that an availability and scalability requirements document is created. The emphasis is on being able to perform the following steps:
 - Gather the IT infrastructure information
 - Gather the network topology information
 - Gather the enterprise data and application information
 - Analyze the gathered information
 - Produce hardware recommendations

- Produce the middleware configuration recommendations
- Document the identity management availability and scalability recommendations
- ▶ Given the existing organization and IT environment, gather the user interface requirements and develop the solution so that a self-service user interface design is created. The emphasis is on being able to perform the following steps:
 - Gather the user activity requirements
 - Gather the interface customization requirements
 - Identify activities to be grouped together
 - Document the interface design
- ▶ Given the component and server layout within the various security zones, identify the transport channels and select their protection methodology, identify the components and their security needs, and design a comprehensive security solution so that a plan to protect Tivoli Identity Manager data as it is stored and transported in and between the various components is created. The emphasis is on being able to perform the following steps:
 - Identify the transport channels
 - Select a channel protection methodology
 - Select a component protection methodology
 - Document the security design
- ▶ Given the architecture design document, create a Tivoli Identity Manager acceptance test strategy so that the delivered result can be validated. The emphasis is on being able to perform the following steps:
 - Define the test phases and scope
 - Gather the requirements for testing the components
 - Define the testing objectives and requirements
 - Analyze the risk assessment
 - Define the testing levels, types, and phases
 - Document the criteria and acceptance test steps

Section 2: Installation

The section provides further information about the installation area of the test:

- ▶ Given the prerequisite and patch software, install and configure prerequisite software so that it is ready for Tivoli Identity Manager. The emphasis is on being able to perform the following steps:
 - Gather the hardware and platform specifications
 - Validate and update the hardware to Tivoli Identity Manager specifications
 - Determine the prerequisite the patch level that is required for Tivoli Identity Manager

- Install the prerequisite patches
- Configure the prerequisites for Tivoli Identity Manager installation
- Verify that the installation and configuration are successful
- ▶ Given the Tivoli Identity Manager Server software and access to the Information Center, review the installation guides and install the software so that the Tivoli Identity Manager Server passes a basic functionality test. The emphasis is on being able to perform the following steps:
 - Review the installation documentation
 - Gather the environment data
 - Install the software
 - Verify that the installation is successful
- ▶ Given the Tivoli Identity Manager adapter software, install the adapter on the managed resource and the adapter profile on the Tivoli Identity Manager Server so that the adapter is properly installed and functioning. The emphasis is on being able to perform the following steps:
 - Install the adapter software
 - Install the profile on Tivoli Identity Manager Server
 - Configure the adapter
 - Verify the installation and configuration are successful
- ▶ Given the installed adapter, create a certificate signing request and install the certificate such that the adapter functions properly with its certificate. The emphasis is on being able to perform the following steps:
 - Gather the information that is required for certificate signing request
 - Create the certificate signing request
 - Install the certificate
 - Test the communication
- ▶ Given an installed Tivoli Identity Manager application and a test plan, log in and use the system functions to validate that Tivoli Identity Manager is running properly. The emphasis is on being able to perform the following steps:
 - Start the Tivoli Identity Manager environment
 - Review the logs to ensure clean startup
 - Execute the test plan and verify success
 - Document the results

- ▶ Given the Tivoli Directory Integrator software, functioning Tivoli Identity Manager Server and the server which Tivoli Directory Integrator will be installed on, install and configure Tivoli Directory Integrator server so that the Tivoli Directory Integrator server is running properly. The emphasis is on being able to perform the following steps:
 - Review the installation documentation
 - Gather the environment data
 - Install the software, including the latest fixpack
 - Verify that the installation is successful

Section 3: Implementation

The section provides further information about the implementation area of the test:

- ▶ Given a newly installed Tivoli Identity Manager Server, evaluate and configure the environment values so that the Tivoli Identity Manager Server settings are optimally configured. The emphasis is on being able to perform the following steps:
 - Document the initial settings for the Tivoli Identity Manager Server application, WebSphere Application Server, Tivoli Identity Manager HTTP Server, Tivoli Identity Manager database, and Tivoli Identity Manager directory server.
 - Refer to the *IBM Tivoli Identity Manager Version 5.0 Performance Tuning Guide, SC23-6594* for recommended initial configuration settings for each component.
 - Set the initial configuration parameters for each component.
 - Document the new configuration settings for each component.
- ▶ Given the appropriate organizational design documents, create the required organizational containers such that the organization structure is configured. The emphasis is on being able to perform the following steps:
 - Create any additional organizations
 - Create the organizational units
 - Create the locations
 - Create the business partner organizations
 - Create the administration domains
- ▶ Given the appropriate Tivoli Identity Manager group and ACI design and access to the Tivoli Identity Manager GUI, create the Tivoli Identity Manager groups, ACIs, and relationship expressions such that the Tivoli Identity Manager security model meets customer expectations. The emphasis is on being able to perform the following steps:
 - Create groups
 - Create the organizational ACIs

- Create the provisioning ACIs
- Create the report ACIs
- Create the category ACIs
- Create the required LDAP indexes for attributes that are defined in relationship expressions
- ▶ Given the object classes, an appropriate list of attributes, and access to the LDAP tool, create and configure custom attributes such that the schema is extended. The emphasis is on being able to perform the following steps:
 - Add the attributes to LDAP
 - Create the new custom classes
 - Create custom labels
 - Add attributes to the service schema
 - Add attributes to the adapter schema
 - Add indexes as required
- ▶ Given the entities design, create custom entities that satisfy customer requirements. The emphasis is on being able to perform the following steps:
 - Add the entity
 - Configure a default search attribute
 - Configure a name attribute
 - Configure mapped attributes
 - Create custom operation definitions
 - Save the entity
- ▶ Given the appropriate forms design, configure the forms such that all required forms meet the design requirements. The emphasis is on being able to perform the following steps:
 - Select the form to customize
 - Add or remove tabs
 - Add or remove attributes
 - Change control types
 - Populate the attribute lists
 - Configure the attribute parameters
 - Save the form template
- ▶ Given the appropriate organizational roles design, create the static or dynamic roles, or both, such that they are configured. The emphasis is on being able to perform the following steps:
 - Create static roles
 - Create dynamic roles, including an LDAP filter

- ▶ Given the appropriate services design and managed services data, create Tivoli Identity Manager service objects such that they are configured and functioning. The emphasis is on being able to perform the following steps:
 - Determine the service type, including manual services
 - Populate the service form
 - Test the service connectivity
 - Save the service
 - Set the policy enforcement type
 - Configure the compliance alert method
- ▶ Given the appropriate workflow design and custom workflow extensions, create workflows such that they satisfy customer requirements. The emphasis is on being able to perform the following steps:
 - Determine the workflow type
 - Define the workflow data
 - Add elements to the workflow
 - Configure the elements
 - Connect the elements
 - Configure the notification templates
 - Configure the action text
 - Save the workflow
- ▶ Given the appropriate service selection policy design and the JavaScript extensions, enter the definition for each service selection policy so that they function as required. The emphasis is on being able to perform the following steps:
 - Populate the general information
 - Determine the service type
 - Enter the JavaScript definition
 - Save the changes to policy
- ▶ Given the appropriate provisioning policy design, add entitlements, memberships, and targets such that the provisioning policies are properly configured. The emphasis is on being able to perform the following steps:
 - Populate the general information
 - Add memberships
 - Add entitlement
 - Set the target type
 - Configure the parameter lists
 - Associate the workflow
 - Save changes to the policy

- ▶ Given the appropriate join directives design and custom join directive extension, set the join directives for each profile so that the join directives are set. The emphasis is on being able to perform the following steps:
 - Select the service profile
 - Select the attribute
 - Set the join type
 - Save the join directives
- ▶ Given the password policy design and custom password policy extension, create the password policy such that it creates the appropriate passwords for the specified service type. The emphasis is on being able to perform the following steps:
 - Copy the custom password policy extension files
 - Edit the password policies file
 - Edit the custom labels file
 - Restart the Tivoli Identity Manager Server
 - Populate the general information
 - Choose the target service types or instances
 - Set the password rules
 - Save the policy changes
- ▶ Given the identity policy design, create an identity policy such that it creates the appropriate IDs for the specified service type. The emphasis is on being able to perform the following steps:
 - Populate the general information
 - Choose the target service types or instances
 - Enter the JavaScript definition
 - Save the policy changes
- ▶ Given the password configuration design, configure the password settings such that passwords are handled appropriately throughout Tivoli Identity Manager. The emphasis is on being able to perform the following steps:
 - Configure the lost password behavior
 - Configure the challenge and response settings
 - Enable or disable password editing
 - Enable or disable password synchronization
 - Set the password expiration period
 - Set the password retrieval period
 - Set the maximum number of invalid login attempts
 - Save the password settings

- ▶ Given the appropriate user interface parameters design and access to the ui.properties file, configure the ui.properties file so that the user interface requirements meet customer expectations. The emphasis is on being able to perform the following steps:
 - Configure the customer logo
 - Configure the page size
 - Configure the page link maximum
 - Configure the search results maximum
 - Configure the console title bar
 - Configure the console banner
 - Configure the console footer
 - Configure the post office template size limits
 - Configure the report limits
- ▶ Given the appropriate installation and custom files, configure e-mail properties for password notification so that the settings are configured. The emphasis is on being able to perform the following steps:
 - Configure the password notification method
 - Configure the property files
 - Add custom password notification workflows
- ▶ Given an e-mail management design, configure the Tivoli Identity Manager post office settings such that the e-mail management requirements have been met. The emphasis is on being able to perform the following steps:
 - Configure the system-wide post office setting
 - Configure the collection interval
 - Configure the post office settings on the manual activity nodes
 - Define the aggregate message
- ▶ Given an e-mail management design, configure the workflow notification templates such that the e-mail management requirements have been met. The emphasis is on being able to perform the following steps:
 - Configure the default escalation limit
 - Configure the reminder intervals
 - Customize the default notification templates
- ▶ Given the default e-mail notification template, perform the modifications, test, and implement steps so that the e-mail notification contains the requested information that can be shared across multiple workflows. The emphasis is on being able to perform the following steps:
 - Clone the default template
 - Define the subject and body
 - Determine the xhtml content
 - Include the notification in a workflow

- ▶ Given the standard self-service view, perform the customizations so that customer requirements are met. The emphasis is on being able to perform the following steps:
 - Locate the default views
 - Modify tasks content that is available for a specified view
 - Create the ACIs for view content
 - Modify operations to enable a task in the view
 - Re-order tasks on the homepage
 - Control the page layout
 - Modify the content for custom specifications
 - Customize the style sheets to match a corporate specification
- ▶ Given the appropriate self-service interface parameters and the SelfServiceUI.properties file, configure the SelfServiceUI.properties file so that the self-service interface satisfies the customer requirements. The emphasis is on being able to perform the following steps:
 - Configure the page size
 - Configure the page link maximum
 - Configure the search results maximum
 - Configure the layout options
 - Configure the user search attributes

Section 4: Data management

The section provides further information about the data management area of the test:

- ▶ Given the detailed design, human resources data, and the Tivoli Identity Manager schema, determine the identity data sources and the load method to create an identity loading process. The emphasis is on being able to perform the following steps:
 - Identify the data sources
 - Determine the load method (Tivoli Directory Integrator, JNDI, DSML, LDAP, AD)
 - Map the external data to the Tivoli Identity Manager schema
 - If using Tivoli Directory Integrator, configure the Tivoli Directory Integrator server AssemblyLine
- ▶ Given a human resources feed data file and the organization structure design, create a Tivoli Identity Manager HR feed service, schedule, and run a reconciliation such that the data is loaded correctly into the Tivoli Identity Manager repository. The emphasis is on being able to perform the following steps:
 - Create the HR Feed service (DSML, Tivoli Directory Integrator, AD, LDAP)
 - Define a placement rule

- Schedule the reconciliation
- Initiate the reconciliation
- Validate the reconciled user data
- ▶ Given the Tivoli Identity Manager adapter and service definition, migrate existing accounts so that the accounts are associated with the appropriate identities. The emphasis is on being able to perform the following steps:
 - Define the reconciliation for services
 - Define the adoption rules at the appropriate level
 - Run the initial reconciliation
 - Verify the reconciliation results
- ▶ Given the orphan accounts and their appropriate owners, configure the correct owner's person records so that the orphan accounts are adopted. The emphasis is on being able to perform the following steps:
 - Identify orphan accounts
 - Identify owners for orphan accounts
 - Map the account to the owner using JavaScript or the preferred user ID
 - Define a method to manage system accounts
 - Run the reconciliation again
 - Verify that orphans get adopted by the correct person records

Section 5: Troubleshooting

The section provides further information about the troubleshooting area of the test:

- ▶ Given access to the relevant logs and files, review logs so that the issues are identified. The emphasis is on being able to perform the following steps:
 - Gather log files
 - Review the Tivoli Identity Manager log files
 - Review middleware logs (IBM DB2, IBM Tivoli Directory Server, IBM WebSphere Application Server)
 - Determine the problem category
 - Increase the logging level for the appropriate category
 - Reproduce the problem if possible
- ▶ Given a problem description, analyze the data flow so that the component that is the source of the problem is isolated. The emphasis is on being able to perform the following steps:
 - Determine the source of the data
 - Determine all that components that store or move the data
 - Isolate the components that perform operations on the data
 - Analyze logs and audit records to verify data integrity at all steps
 - Identify components where the data is mishandled

- ▶ Given adapter-related problems, troubleshoot the source of the problem so that the problem is identified. The emphasis is on being able to perform the following steps:
 - Analyze the completed or pending requests view
 - Gather log data from adapter and server
 - Analyze the log data and audit records
 - Modify the server and adapter logging levels as necessary

Section 6: Production

The section provides further information about the production area of the test:

- ▶ Given a functioning test environment and production systems, copy configurations to the production environment such that the production system mirrors the test systems and functions with production adapters. The emphasis is on being able to perform the following steps:
 - Enable security on the production system
 - Promote customizations from test to production using appropriate tools
 - Promote Tivoli Identity Manager configuration data to production using appropriate tools
 - Modify services to match the production adapters
 - Reconcile the supporting data production adapters
 - Test the production system
- ▶ Given a list of services and a schedule for the reconciliations, create reconciliation schedules for each service with appropriate filters. The emphasis is on being able to perform the following steps:
 - Determine the systems to reconcile
 - Determine the frequency of reconciliation for each service
 - Define any reconciliation filters for each service
 - Create a reconciliation schedule for each service
- ▶ Given a production environment copied from a functioning test environment and the acceptance test plan, perform production verification and acceptance so that the production system is functional. The emphasis is on being able to perform the following steps:
 - Execute the test plan
 - Validate the communication between the Tivoli Identity Manager Server and all adapters
 - Validate the provisioning policy changes using policy preview
 - Validate that e-mail notifications are reaching the appropriate target or targets
 - Validate the user interface

- ▶ Given the existing security strategy and SSL certificates, install the certificates and enable SSL on all components so that secure communication between Tivoli Identity Manager and the middleware and adapters is configured. The emphasis is on being able to perform the following steps:
 - Configure the Tivoli Identity Manager HTTP Server for HTTPS only communications with the user and install the certificate
 - Install the certificates in WebSphere
 - Install the certificates in ADK adapters
 - Install the certificates in Tivoli Directory Intergrator
 - Enable SSL on ADK adapters and Tivoli Directory Intergrator
 - Install the certificates on LDAP server
 - Configure the LDAP server to use SSL
 - Configure Tivoli Identity Manager to use SSL for LDAP connections

Section 7: Maintenance

The section provides further information about the maintenance area of the test:

- ▶ Given the Tivoli Identity Manager systems, implement monitoring procedures so that the Tivoli Identity Manager deployment can be monitored. The emphasis is on being able to perform the following steps:
 - Monitor the connectivity to database, LDAP, and adapters
 - Monitor the disk space of application servers and repositories
 - Track logs and log sizes
 - Monitor error logs for problems
 - Manage the LDAP recycle bin if enabled
 - Monitor cluster members
 - Schedule system backups
 - Monitor performance
- ▶ Given the Tivoli Identity Manager upgrade software and documentation, upgrade Tivoli Identity Manager on test and production systems so that it is functioning properly. The emphasis is on being able to perform the following steps:
 - Determine the middleware components to upgrade
 - Obtain the server component upgrade software
 - Request backup of all Tivoli Identity Manager components
 - Request backup of all system components
 - Create the test environment
 - Install an upgrade on test environment
 - Repackage the custom applications with the upgraded API .jar files
 - Validate the test upgrade environment
 - Install the upgrade on the production server
 - Test the server

- ▶ Given the adapter software and documentation, upgrade and test the Tivoli Identity Manager adapters so that they are upgraded and functioning properly. The emphasis is on being able to perform the following steps:
 - Obtain the new adapter software
 - Determine the components to install
 - Request to upgrade the backup systems
 - Install the new adapter or upgrade
 - Install the adapter profile
 - Verify the certificates
 - Test the adapter
- ▶ Given the fix pack software and documentation, install the appropriate fix pack on the test and production systems such that the software is functioning properly. The emphasis is on being able to perform the following steps:
 - Obtain the fix pack software
 - Determine the requirements for the fix pack from the fix pack documents
 - Create a test environment
 - Install the fix pack on the test environment
 - Perform the function test on items that are fixed by fix pack
 - Request a backup of the system
 - Install the fix pack on the production server
 - Test the fix pack
- ▶ Given the *IBM Tivoli Identity Manager Version 5.0 Performance Tuning Guide*, SC32-6594, and the customer's hardware specifications, configure system settings such that Tivoli Identity Manager is tuned and functioning properly. The emphasis is on being able to perform the following steps:
 - Identify Tivoli Identity Manager deployment parameters and settings
 - Use the *IBM Tivoli Identity Manager Version 5.0 Performance Tuning Guide*, SC32-6594
 - Set the memory settings
 - Configure the logging levels, options, and file sizes
 - Set messaging and timeout values
 - Set disk usage limits
- ▶ Given workload information and archive requirements, configure and schedule directory and database cleanup so that historical and temporary objects are removed. The emphasis is on being able to perform the following steps:
 - Enable the recycle bin
 - Configure recycle bin age limit
 - Create a cron job for recycle bin cleaning
 - Perform a database backup
 - Create the SQL command for database cleaning

Section 8: Enhancements in V5.0

The section provides further information about the enhancements in Tivoli Identity Manager V5.0:

- ▶ Given an access plan and the target resource, define and validate the access entitlements for each participant so that it is verified that the access entitlements are configured correctly for the resource. The emphasis is on being able to perform the following steps:
 - Select the participants
 - Define the access entitlements
 - Validate the access entitlements
- ▶ Given the security and compliance requirements and a deployed Identity Management solution, create and schedule a recertification policy so that a recertification policy is created. The emphasis is on being able to perform the following steps:
 - Define the general parameters
 - Choose the type and target
 - Set the schedule for execution
 - Define the policy actions
 - Select or customize the e-mail notifications
 - Define customizations to the workflow
- ▶ Given the appropriate self-service user interface design, configure the self-service user interface so that it meets customer requirements. The emphasis is on being able to perform the following steps:
 - Configure the main page layout by modifying the SelfServiceUI.properties file
 - Configure the screen text by modifying the SelfServiceScreenText_.properties file
 - Customize the Web content by modifying the files in the itim_self_service.war/custom directory
 - Customize the help content by modifying the SelfServiceHelp.properties file
 - Configure the default home page properties by modifying the SelfServiceHomePage.properties file
 - Define the views from the self-service UI configuration page
- ▶ Given the proper documentation and extension jar file, configure a new JavaScript extension so that the JavaScript extension satisfies the customization requirements. The emphasis is on being able to perform the following steps:
 - Gather the extension documentation
 - Define the required Tivoli Identity Manager modules that are affected
 - Modify the scriptframework.properties file as required

- Modify the WebSphere properties as required
- Document the modifications to current system
- ▶ Given the existing reporting requirements and the Tivoli Identity Manager report pack, install the Tivoli Common Reporting Server so that reports are created. The emphasis is on being able to perform the following steps:
 - Install the Tivoli Common Reporting Server
 - Deploy the Tivoli Identity Manager report pack
 - Customize the Tivoli Identity Manager reports as necessary
 - Run the reports and verify their accuracy

1.3 Recommended educational resources

Courses and publications are offered to help you prepare for the certification tests. The courses are recommended, but not required, before taking a certification test. If you want to purchase Web-based training courses or cannot locate a Web-based course or classroom course at the time and location you desire, contact one of our delivery management teams at:

- ▶ Americas:
<mailto:tivamedu@us.ibm.com>
- ▶ EMEA:
<mailto:tived@uk.ibm.com>
- ▶ Asia-Pacific:
<mailto:tivtrainingap@au1.ibm.com>

Note: Course offerings are continuously added and updated. If you do not see courses listed in your geographical location, contact the delivery management team.

1.3.1 Courses

This section provides information about the currently available or planned Tivoli Identity Manager V5.0 courses. Refer to the Tivoli software education Web site to find the appropriate courses and education delivery vendor for each geography, available at:

<http://www.ibm.com/software/tivoli/education>

General training information is available at:

<http://ibm.com/training>

You can also refer to the existing *Administrator skills roadmap for IBM Tivoli Identity Manager 5.0* and *Implementor skills roadmap for IBM Tivoli Identity Manager 5.0* at the following Web site:

ftp://ftp.software.ibm.com/software/tivoli/education/Roadmaps/ITIM_50.pdf

IBM Tivoli Identity Manager V5.0 Introduction

This course introduces the basic concepts of Tivoli Identity Manager 5.0. The students will broaden their knowledge and understanding of Tivoli Identity Manager fundamentals such as value add concepts, provisioning, management of personal information, and workflows. Additionally, students will learn about the components used in Tivoli Identity Manager and the basic architecture of a typical environment.

Course duration

This is a two-hour, self-paced course.

Objectives

After taking this course, you will be able to:

- ▶ Describe the value proposition of Tivoli Identity Manager V5.0.
- ▶ Describe policy-based provisioning.
- ▶ Describe Tivoli Identity Manager functionality, provisioning policies, and workflows to support various types of provisioning scenarios.
- ▶ Explain how to use Tivoli Identity Manager to grant access.
- ▶ Describe how to use Tivoli Identity Manager V5.0 to manage personal information, organizations, and people, and how to provision accounts.
- ▶ Identify the major components of the Tivoli Identity Manager V5.0 system architecture.

Outline

The course follows this outline:

1. Users, Services, and Accounts:
 - Introduction to Tivoli Identity Manager
 - Identify the types of users in a large environment
 - Define services, roles, and accounts as used in Tivoli Identity Manager
 - Describe how to use Tivoli Identity Manager to manage personal and organizational information

2. Provisioning:
 - Define Tivoli Identity Manager policies and workflows.
 - Describe policy-based provisioning.
 - Describe how to use Tivoli Identity Manager to provision accounts.
3. Tivoli Identity Manager Architecture:
 - Identify the major components of the Tivoli Identity Manager V5.0 system architecture.
 - Explain how services receive accounts from Tivoli Identity Manager.

Required skills

Before taking this course, you should possess the following knowledge and skills:

- ▶ Basic system configuration and management concepts
- ▶ Database servers
- ▶ Directory server and LDAP concepts
- ▶ IT security policy concepts

IBM Tivoli Identity Manager 5.0 Differences and Migration

This course covers differences between the new Tivoli Identity Manager V5.0 product and the last version, Tivoli Identity Manager V4.6. The differences covered in this course are mainly related to the operations that are performed most often by administrators and users, including, but not limited to, creating services, creating provisioning policies, and requesting access. Additionally, new features, such as the self service console, a new auditor role and related reports, access entitlements, and migration from a previous version of Tivoli Identity Manager will be covered.

Course duration

This is a three-day, classroom course.

Objectives

After taking this course, you will be able to:

- ▶ Identify key new features in Tivoli Identity Manager V5.0
- ▶ Create and manage access entitlements
- ▶ Request access using the self service console
- ▶ Audit request activities
- ▶ Perform a migration from Tivoli Identity Manager V4.6 to Tivoli Identity Manager V5.0

Outline

The course follows this outline:

- ▶ Key New Features
- ▶ Identity Feeds
- ▶ Adapters and Services
- ▶ Policies
- ▶ Workflow Enhancements
- ▶ Access Entitlements
- ▶ Self Service
- ▶ Auditing and Reporting
- ▶ Migration

Required skills

The following skills are required for this course:

- ▶ IBM WebSphere
- ▶ DB2
- ▶ IBM Tivoli Directory Server
- ▶ IBM Tivoli Directory Integrator
- ▶ IBM Tivoli Identity Manager
- ▶ Linux® system administration

IBM Tivoli Identity Manager V5.0 Basic Implementation

This course covers basic implementation of Tivoli Identity Manager 5.0. You will install and configure a typical Tivoli Identity Manager 5 deployment on Linux. This five-day course includes intensive hands-on exercises in addition to the lecture.

Course duration

This is a five-day, classroom course.

Objectives

After taking this course, you will be able to:

- ▶ Describe the process of creating a needs-assessment document for Tivoli Identity Manager
- ▶ Describe the value proposition of Tivoli Identity Manager 5.0
- ▶ Identify the major components of the Tivoli Identity Manager 5.0 system architecture
- ▶ Install Tivoli Identity Manager 5.0 and prerequisite middleware and any required fixpacks on Linux
- ▶ Configure organizational units, locations, and administrative domains
- ▶ Navigate through LDAP directory data

- ▶ Create static and dynamic organizational roles
- ▶ Create multiple identity feeds
- ▶ Create services
- ▶ Create identity, password, and service selection policies
- ▶ Configure scheduled and manual reconciliations
- ▶ Manually request, modify, suspend, restore and delete accounts
- ▶ Describe workflow elements and create basic workflows
- ▶ Describe Notifications and use the Notification Post Office
- ▶ Create groups, views and Access Control Items (ACIs)
- ▶ Create and manage entitlements
- ▶ Create provisioning policies and set join directives
- ▶ Correct noncompliant accounts and adopt orphaned accounts
- ▶ Manage account request activities
- ▶ Describe and configure life cycle management
- ▶ Use the forms customization applet
- ▶ Configure service forms, password settings and synchronization
- ▶ Customize the Administration and Self Service Consoles
- ▶ Generate reports and design custom reports
- ▶ Audit request activities
- ▶ Describe methods of problem determination

Outline

The course follows this outline:

- ▶ Introduction
- ▶ Planning
- ▶ Installation
- ▶ Organization Management
- ▶ User Management and Identity Feeds
- ▶ Access Control
- ▶ Services and Policies
- ▶ Provisioning
- ▶ Entitlement Workflows

- ▶ Life Cycle Management
- ▶ Auditing and Reporting
- ▶ Customization
- ▶ Problem Determination

Required skills

The following skills are required for this course:

- ▶ Familiarity with LDAP and TCP/IP fundamentals
- ▶ Familiarity with JavaScript
- ▶ Basic administrative skills for:
 - Linux
 - IBM Tivoli Directory Server
 - IBM WebSphere Application Server

IBM Tivoli Identity Manager V5.0 System Administration

The Tivoli Identity Manager V5.0 System Administration course is catered to Tivoli Identity Manager system administrators. Materials covered includes: system architecture, creating services, creating policies, performing reconciliations, and loading data. This course is comprised of lecture and labs.

Course duration

This is a three-day, classroom course.

Objectives

After taking this course, you will be able to:

- ▶ Ensure the health of the Tivoli Identity Manager implementation and middleware
- ▶ Apply fixes to Tivoli Identity Manager and middleware
- ▶ Upgrade Tivoli Identity Manager components

Outline

The course follows this outline:

- ▶ Maintaining System Health
 - Set memory settings (Lab)
 - Monitor connectivity to database and LDAP (Lab)
 - Monitor disk space of application servers and repositories (Lab)
 - Track logs and log sizes (Lab)
 - Monitor error logs for problems (Lab)
 - Manage LDAP recycle bin (Lab)

- Monitor cluster members (Lab)
- Schedule system backups
- Configure logging levels, options and file sizes (Lab)
- Set messaging - timeout values (Lab)
- Set disk usage limits (Lab)
- Set thread count
- Configure recycle bin age limit (Lab)
- ▶ Problem Determination and management
 - Gather log files (Lab)
 - Review Tivoli Identity Manager log files
 - Review middleware logs (DB2, IDS, WebSphere Application Server)
 - Determine problem category (Lab)
 - Increase logging level for appropriate category (Lab)
 - Reproduce problem if possible
 - Take appropriate action
 - Given a non-working configuration and access to relevant logs and files, identify the source of problems (Lab)
 - Given a problematic data management scenario, identify the source of problems (Lab)
 - Given adapter related problems, troubleshoot to identify the source of the problem and resolve if possible (Lab)
- ▶ Upgrading and Applying System Fixes
 - Planning the fix cycle - Production to Test to Production
 - Moving from Test to Production
 - Install upgrade on production server
 - Promote Tivoli Identity Manager configuration data to production using appropriate tools
 - Promote customizations from test to production using appropriate tools
 - Modify services to match production adapters
 - Reconcile production adapters
 - Test the production system
 - Applying Changes to Test Environment
 - Request backup of all Tivoli Identity Manager components
 - Request backup of all system components
 - Create Test environment
 - Install upgrade on Test environment (Lab)
 - Validate “Test” upgrade environment

Required skills

The following skills are required for this course:

- ▶ Basic operating-system administrative skills for UNIX® and Windows®
- ▶ LDAP experience
- ▶ TCP/IP fundamentals
- ▶ Firewall concepts
- ▶ Working knowledge of web protocols (HTTP, XML)
- ▶ Experience in reading, interpreting, and creating regular expressions
- ▶ Experience in modifying system parameters
- ▶ JavaScript writing and reading

1.3.2 Publications

Tivoli Identity Manager guides and IBM Redbooks publications are useful tools for preparing to take Test 934.

IBM Tivoli Identity Manager product documentation

You might want to refer to the following guides:

- ▶ IBM Tivoli Identity Manager Information Center
<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>
- ▶ *IBM Tivoli Identity Manager Server Version 5.0 Installation and Configuration Guide*, SC32-1562
- ▶ *IBM Tivoli Identity Manager Version 5.0 Database and Schema Reference*, SC32-9011
- ▶ *IBM Tivoli Identity Manager Version 5.0 Problem Determination Guide*, SC32-1561

Technical supplement

- ▶ *IBM Tivoli Identity Manager Version 5.0 Performance Tuning Guide*, SC32-6594

IBM Redbooks publications

Refer to the following books on topics that are related to Tivoli Identity Manager:

- ▶ *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014

This IBM Redbooks publication reviews the overall Tivoli Enterprise Security Architecture. It focuses on the integration of audit and compliance, access control, identity management, and federation throughout extensive

e-business enterprise implementations. The available security product diversity in the marketplace challenges everyone in charge of designing single secure solutions or an overall enterprise security architecture. With Access Manager, Tivoli Identity Manager, Federated Tivoli Identity Manager, Security Compliance Manager, Security Operations Manager, Directory Server, and Tivoli Directory Integrator, Tivoli offers a complete set of products designed to address these challenges.

This book describes the major logical and physical components of each of the Tivoli products. It also depicts several e-business scenarios with different Tivoli security challenges and requirements. By matching the desired Tivoli security product criteria, this publication describes the appropriate security implementations that meet the targeted requirements.

This book is a valuable resource for security officers, administrators, and architects who want to understand and implement enterprise security following architectural guidelines.

- ▶ *Integrated Identity Management using IBM Tivoli Security Solutions*, SG24-6054

This IBM Redbooks publication provides a solution-oriented overview of using Tivoli security products to provide an implementation for integrated identity management based on real-life customer experience.

When defining functional requirements for e-business-related projects, you have to take into consideration a serious amount of security-related tasks and disciplines. These disciplines are authentication and credential acquisition, use of directory infrastructures, session management, multiple tiers of single sign-on, authorization, administration, users and policy, accountability, and availability. Together they stand for the integrated identity management approach, an approach that should be regarded as a holistic way of tying security requirements into your projects.

This book is a valuable resource for security officers, administrators, and architects who want to understand and implement enterprise security following these guidelines.

- ▶ *Identity Management Design Guide using IBM Tivoli Identity Manager*, SG24-6996

Identity management is the concept of providing a unifying interface to manage all aspects related to individuals and their interactions with the business. It is the process that enables business initiatives by efficiently managing the user life cycle, including identity/resource provisioning for people (users), and by integrating into the required business processes. Identity management encompasses all the data and processes related to the representation of an individual involved in electronic transactions.

This IBM Redbooks publication provides a methodology for designing an identity management solution with Tivoli Identity Manager V4.6. Starting from a high-level, organizational viewpoint, we show how to define user registration and maintenance processes using the self-registration and self-care interfaces and the delegated administration capabilities. Using the integrated workflow, we automate the submission and approval processes for identity management requests, and with the automated user provisioning, we take workflow output and automatically implement the administrative requests on the environment with no administrative intervention.

This book is a valuable resource for security administrators and architects who want to understand and implement a centralized identity management and security infrastructure.

- ▶ *Identity Management Advanced Design for IBM Tivoli Identity Manager, SG24-7242*

Identity and user life cycle management projects are being deployed more and more frequently—and demand is growing. By demonstrating how Tivoli Identity Manager can be made resilient and adapted to special functional requirements, this IBM Redbooks publication creates or enhances confidence in the Tivoli Identity Manager-based solution for senior management, architects, and security administrators.

Advanced design topics can start with infrastructure availability for all involved components, Web application, and database server clustering as well as LDAP multi-master setups, continuing with compliance challenges addressing enhanced auditing and reporting, and designing and creating your own self-care/self-registration application environment that embraces external users and business partners offering fine-tuned workflow options and life cycle management capabilities.

The powerful features and extensions of Tivoli Identity Manager are opening doors into a world of advanced design and customization for every identity management challenge you might encounter.

- ▶ *Deployment Guide Series: IBM Tivoli Identity Manager 5.0, SG24-6477*

Deploying an identity management solution for a medium size business begins with a thorough analysis of the existing business and IT environment. After we fully understand the organization, their deployed infrastructure, and the application framework, we can define an applicable representation of these assets within an identity management implementation.

This IBM Redbooks publication takes a step-by-step approach to implementing an identity management solution based on Tivoli Identity Manager. Part 1 takes you through an example company profile with existing business policies and guidelines and builds an identity management solution design for this particular environment. In Part 2, describes how the new

identity management components can be integrated into the existing environment. Then, it focuses on the detailed configuration of identity management integration tasks that must be implemented to create a fully functional end-to-end solution.

This book does not introduce any general identity management concepts, nor does it systematically explain all of the Tivoli Identity Manager components and capabilities; instead, those details are thoroughly discussed in *Identity Management Design Guide using IBM Tivoli Identity Manager*, SG24-6996, and *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014.

Archived

Archived

Planning

In this chapter, we discuss the aspects of planning an IBM Tivoli Identity Manager solution. The following high-level steps are required:

- ▶ Understanding the customer environment and business processes
- ▶ Gathering the requirements for the identity management solution
- ▶ Designing the identity management solution
- ▶ Documenting the solution

When professionally planning for your deployment, also consult *IBM Tivoli Identity Manager Version 5.0 Performance Tuning Guide*, SC32-6594. Performance tuning is a discipline that is best applied in the planning stage as well as after the fact. Another great source for Tivoli Identity Manager planning (and other) related topics is the Tivoli Identity Manager wiki, which is available at:

<http://www.ibm.com/developerworks/wikis/display/tivoliim/Home>

2.1 Overview

The first step when you plan an identity management project is to define its *scope*, which involves details about the current environment, the goal of the solution, and its functional and technical requirements.

The initial project definition is usually based on the business and functional requirements that triggered this project along with documentation, such as the IT architecture, security architecture, or equivalent. These documents identify the business background, the business need for the solution, and, typically, the business and technical requirements for the solution. For an identity management solution, you need to define the following areas in this phase (in no particular order):

- ▶ **User management procedures:** The procedures for managing users, who manages users, and what is required of the new solution for managing users. Also important at this stage is to understand what the authoritative source for user data is; most likely, it is closely connected to the human resources department.
- ▶ **Password management procedures:** The procedures for managing account passwords, who manages passwords, and what is required of the new solution for managing passwords.
- ▶ **Access control management procedures:** The procedures for managing access control, who manages the access control definition, and what is required for the new solution for managing access control.
- ▶ **Security policy:** What the corporate security policy defines for users, accounts, passwords, and access control mechanisms.
- ▶ **Target systems:** The current system environment (including operating systems, databases, applications, the network, firewalls, physical locations, and access control) and the system requirements of the new solution.
- ▶ **Interfaces:** The interfaces to the current identity management mechanisms and procedures, and the integration requirements of the new solution.
- ▶ **Auditing and reporting procedures:** The procedures for auditing and reporting, who is involved in the auditing and reporting of users and their access, and the audit requirements for the new solution.
- ▶ **Technical requirements:** The other technical requirements for the solution, such as availability, scalability, monitoring, and recovery.

Gathering this information normally involves a series of interviews and workshops with the people and teams that are involved in identity management. These terms can include the CIO, IT executives, the security management or administration team, operations, help desk, key technical teams (Microsoft®

Windows admin, UNIX sysadmin, and so on), any application development teams, and business managers who are involved in the project. The combination of these interviews and workshops helps develop a picture of how the system currently works and how it can be improved. It is important to vet the “wish list” from the genuine requirements. The project owners should drive the requirements for the proposed system, although others can contribute to an understanding of the need for the requirements.

A key component of delineating the definition and design phases is that the existing system and solution requirements are agreed upon between the project owner and the project team prior to the design phase.

2.2 Organization structure design

The function of the Tivoli Identity Manager organization tree (or *org tree*) has been changed in Tivoli Identity Manager V5.0. Instead of using an organizational chart for management, Tivoli Identity Manager has a redesigned administrative interface that combines all organizational objects in a given task. Within the task interface, Tivoli Identity Manager provides the ability to search for and assign objects to a business unit.

In this release, you cannot browse and create entities by navigating the organization tree. The association to a business unit within the organization tree is specified during the creation of the entity.

For example, when working with provisioning policies, you create and assign a policy to a specific business unit. When searching for policies, you can choose to search for all policies in all organizations or by business unit.

Org tree is still used to define the structure for the organization into which Tivoli Identity Manager is being deployed. The tree consists of the following components:

- ▶ An organization: This identifies the top of an organizational hierarchy, which might contain subsidiary entities such as organization units, business partner organization units, and locations. This is the parent node at the top of the node tree. It is possible to create and manage multiple organizations within Tivoli Identity Manager system.
- ▶ One or more locations: These are locations that are defined by the business.
- ▶ One or more organizational units: These are teams or departments as defined by the business.

- ▶ One or more business partner organization units: These are business partners as defined by the business.
- ▶ One or more admin domains: These are Tivoli Identity Manager groupings for administration.

There is no technical difference between locations, organizational units, or business partner organizations. They use different icons and allow the org tree to be modelled as the administrators see fit.

All people are attached to the org tree at a single point.

A policy is attached to points in the org tree and can apply to objects at that level or to all objects at or below that level. This policy can control the provisioning of accounts, account user ID generation, and password strength. Therefore, you can have a corporate-wide password policy defined at the organization level in the org tree or a specific password policy that applies to a specific branch or department of the organization.

Tivoli Identity Manager roles and access control items (ACIs) are also attached to nodes in the org tree, which define the scope of specific access rights within the Tivoli Identity Manager product.

Organizational roles are used to model job roles within an organization. They can be used to map users to a set of accounts that are granted through a provisioning policy.

Delegated administration

The design of the organization tree is impacted by the way you use access control items (ACIs) to set up administrative roles, the scope of the roles, and who will belong to the roles. Try to keep the administrative structure from becoming too complex.

You can use admin domains to establish the scope of access control for people, policies, and services. On a broad scale, administrative domains can be used in two distinctive ways, which are not mutually exclusive and can be used together:

- ▶ Manage access to services and provisioning policies

This model separates users from resources in resource access control. Admin domains set direct access controls on the services and provisioning policies that map users to resources. Changes to organizational roles that group users according to their resource needs are outside the scope of the administrative domain. This model is suitable for enterprises where access to resources is not separated on a geographical or regional basis.

- ▶ Manage access to organizational units or locations

This model integrates the domain administration control in the organizational structure of the people resources. Organizational access control items and provisioning access control items can be established only for the people in the scope of the domain.

If an organization tree uses admin domains, the individual domain administrators are responsible for the portion of the organization tree that is located below the admin domain container node. However, domain administrators are limited in their actions and cannot perform overall system administration. The level of their access is determined by a set of ACIs.

2.3 Service design

Tivoli Identity Manager manages users on many target systems, including operating systems such as UNIX and Microsoft Windows servers, as well as applications such as databases and business applications.

Tivoli Identity Manager deploys an adapter to perform the administration of accounts on the target systems or applications. Some adapters are deployed to the system or application and interact locally. Other adapters operate remotely and can be deployed anywhere in the network.

The adapter to server communication uses the HTTP protocol. In a production environment, this communication should be secured using SSL (HTTPS communication).

Each adapter instance is defined as a service within the Tivoli Identity Manager Server. Accounts are associated with specific services.

Every type of service has a *service profile*. Some of these profiles are included with the Tivoli Identity Manager installation. For example, there is one service profile for Linux services. The service profile defines the account attributes for that type of service. The profile includes many target specific attributes, and decisions about their use (mandatory, optional, or not in use) are dictated by business needs and requirements.

2.4 Entities design

Tivoli Identity Manager is concerned with managing users and their accounts. Passwords, group memberships, and other attributes are associated with the

users and accounts, which all relate to managed systems and applications. To enable management of users, accounts, and associated information, Tivoli Identity Manager uses an organizational tree and roles, groups and ACIs, and various type of policies. Tivoli Identity Manager also uses elements of workflow, audit logs, and reports. We describe these components in the following sections.

The entities managed by Tivoli Identity Manager are:

- ▶ Users, accounts, and attributes
- ▶ Passwords
- ▶ Group membership
- ▶ Managed systems and applications

2.4.1 Users, accounts, and attributes

A person can be classified as a *person*, a *business partner person* (BPPerson), or *custom person* in Tivoli Identity Manager. A person is typically an employee of the company or organization. A BPPerson is typically an individual who needs access to an organization's resources that are managed by the Tivoli Identity Manager system but who is not considered an employee.

All classes of users are managed in the same way. However, more information is required when adding a person than when adding a BPPerson. A custom person is used when the standard person definition does not suit an organization and has to be extended for the organization.

A person can be placed anywhere in the organization tree, so the organization tree can represent the user structure of a company.

The information used for each person is defined as attributes on the person objects, which can include first, last, and full names, phone numbers, employee number, supervisor, and e-mail address.

The Tivoli Identity Manager person¹ with its attributes is mapped to a directory server *iNetOrgPerson* object with its attributes. To differentiate between a person and a BPPerson, Tivoli Identity Manager maps the BPPerson to an *organizationalPerson* object with its attributes. In a situation where predefined attributes cannot entirely describe a person, new attributes can be defined that will cause changes in the directory server schema. For more details, see Chapter 4, "Implementation" on page 101.

An *account* represents a person's access entitlement to the Tivoli Identity Manager system and to *services* that are managed by Tivoli Identity Manager

¹ The *Tivoli Identity Manager person* refers to the person object as it exists after the base installation of Tivoli Identity Manager.

(managed resource), such as Microsoft Windows, Sun™ Solaris™, SAP®, DB2, Tivoli Access Manager, and so on. Accounts require unique attributes that are defined by the managed resource.

An *orphan account* is an account that is not associated with a person. Orphan accounts are generated when the reconciliation process cannot automatically associate the account with a person using adoption rules.

2.4.2 Passwords

The majority of accounts have passwords associated with them. However, an account that is associated with a manual service type such as Voice mail setup, Telephone setup, Employee badge request, and so on does not have a password defined.

Owners or administrators can manage account passwords centrally using the Tivoli Identity Manager Web interface.

Passwords can be synchronized. The synchronization can be applied to all accounts that are associated with a user or selected accounts. For most passwords, this synchronization is one-way. Tivoli Identity Manager sets the password and pushes it to the managed targets. Tivoli Identity Manager cannot accept a password change request from a target and push this request to all associated accounts. The exception is the Microsoft Windows password synchronization function and the Reverse Password Synchronization for IBM Tivoli Access Manager WebSEAL agent, which both intercept a password change and pass it through Tivoli Identity Manager.

Tivoli Identity Manager can generate a random password that can be displayed to an administrator or mailed to a user. To further improve security, the *shared secret* can be used by an account owner to retrieve a new or changed password for an account when the system is configured to not e-mail passwords in the clear. In that case, the user receives a temporary URL that can be accessed only with the shared secret. In addition, Tivoli Identity Manager can be configured to store the shared secret in LDAP as a hashed value for additional protection.

Tivoli Identity Manager uses a challenge/response function to verify users' identities if they have forgotten their Tivoli Identity Manager password. The user can choose challenge questions from a standard list or can enter challenge questions. When logging in to Tivoli Identity Manager for the first time, the user provides the challenge questions (if configured) and responses. On subsequent logins to Tivoli Identity Manager, the user can select a "Forgot password" option and a subset of the challenge/response questions are used to verify the user's identity.

2.4.3 Group membership

Accounts are granted access to target systems and applications by placing them into groups previously defined on the target systems. These can be groups on UNIX systems or in Windows domains, SAP groups or profiles, or any other access control grouping mechanism.

Group lists, for most managed targets, are updated with the reconciliation function. Therefore, administrators do not manually enter group names; they select from an existing list that is synchronized with the target.

Note: Tivoli Identity Manager does not create or delete groups on managed targets. Also, it does not manage access control lists or resource access on the managed targets. The local administrators or application owners must perform these functions using the native system or application tools.

Service group

With Tivoli Identity Manager, users and administrators can request and manage access to resources such as shared folders, e-mail groups, or applications. Access entitlement can be mapped to an existing group on a managed service.

A *service group* is a new type of entity introduced in Tivoli Identity Manager 5.0. A service group is reconciled from the managed service as part of the supporting data. Administrators can view the groups on a service, manage group members, provide a business friendly name and description of the access represented by the group, and expose the access to a user so that the user can directly request or delete access. Administrators can also assign an admin owner for the access, define approval workflows for access requests, and configure recertification policies for the access based on the group. Business users are no longer required to manage an account directly to gain access to IT resources, because Tivoli Identity Manager integrates account management with access management.

2.4.4 Managed systems and applications

Tivoli Identity Manager manages users on many different managed systems, including operating systems such as UNIX and Windows servers, as well as applications such as databases and business applications.

Tivoli Identity Manager deploys an adapter to perform the administration of accounts on the target systems or applications. Some adapters are deployed physically to the system or application and interact locally. Other adapters operate remotely and can be deployed anywhere in the network.

2.5 Life cycle management design

As we mentioned previously, identity management is the process of managing users and their accounts across all systems. User *life cycle management* is the process through which identities are created, managed, and ultimately destroyed. The life cycle management process is concerned with the following topics:

Identity	The user's credentials, such as user name and password, and information about the user, including name, e-mail address, and phone number.
Access rights	The systems, accounts, and applications to which the user has access and the level of access.
Policy management	Updating of access rights based on membership to a particular group or department and consistent enforcement of corporate policies.
Privacy	Enactment of regulations that require enterprises to secure the privacy of certain types of information that are related to specific individuals.

Life cycle management can be divided into four segments, as depicted in Figure 2-1.

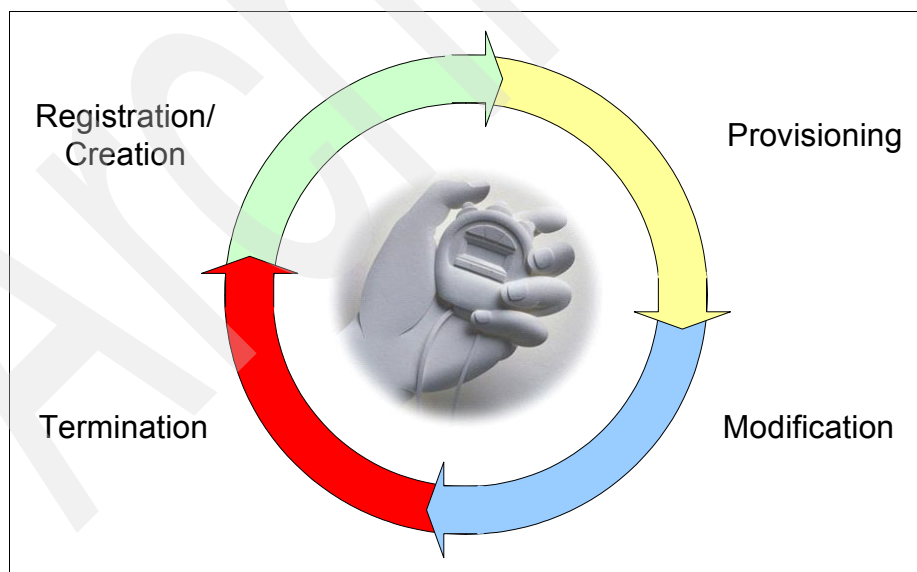


Figure 2-1 Life cycle management overview

In the next section, we take a look at the individual segments.

2.5.1 The registration/creation cycle

The registration/creation cycle begins the process. At this point, we identify a person who needs access to the resources that Tivoli Identity Manager manages. That person needs to be registered/created in Tivoli Identity Manager by describing attributes such as first name, last name, and password.

2.5.2 The provisioning cycle

In the provisioning cycle, the system provides automated process and IT policy enforcement using a coordinated approach to access resources and the associated privileges. Tivoli Identity Manager provides support for *identity provisioning*, the process of providing, deploying, and tracking a user identity in the enterprise.

Provisioning solutions are the link between the classical central management solution and the target resources. The capability to quickly negotiate provisioning requirements that map to the identity models and processes of a business is crucial when architecting a solution. The provisioning aspect garners much of the focus and attention. User provisioning is where the process begins, and if provisioning is sluggish or incomplete, users (employees, consultants, and customers) develop negative first impressions of the organization.

The provisioning cycle includes:

- ▶ Identifying the sponsor (for example, sales or human resources), determining the nature of the relationship (customer or internal employee), verifying the user's identity, and assigning a role or roles to that identity
- ▶ Fulfillment, which entails gaining approval for the appropriate systems, creating the user's identity in the appropriate directories and repositories, and granting access to those accounts

2.5.3 The modification cycle

The modification cycle represents the maintenance phase where created identities experience changes in access rights as the organization changes and as their roles within the organization change. The modification phase of the life cycle offers significant opportunity for automation and efficiency gains.

2.5.4 The termination cycle

Termination is the phase with which, from a security perspective, organizations struggle the most. Auditors discovering hundreds or thousands of user accounts that should have been disabled or deleted is not uncommon.

During the termination phase, organizations need to verify that the relationship between the user and the organization is, in fact, dissolving and disable access accordingly. Often, accounts are disabled for a term and then deleted. Unfortunately, although this sounds simple, it demands process rigor.

2.5.5 Life cycle management

Every person who is registered in Tivoli Identity Manager has a life cycle that Tivoli Identity Manager administrators need to manage, which can be a daunting task if the number of accounts is large, the environment is frequently changed (such as password expiration and account inactivity), or all processing is carried out manually. Administrators can use Tivoli Identity Manager *life cycle rules* to automate the large number of manual tasks that they must perform due to changes in the environment. Life cycle rules are based on business policies that define various system activities, such as the frequency of password changes, the termination of contractors, and standards for defining user ID and password rules on various systems. Life cycle rules can eliminate situations where some policies are not enforced. Administrators can use life cycle rules to define events that are triggered either immediately or at certain time intervals that are defined according to company/business policies.

The *operation workflows* can be used to customize the life cycle management of users and accounts, such as adding, removing, and modifying users and accounts. A complete provisioning workflow system automatically routes requests to the proper approvers and preemptively escalates to alternate approvers if actions are not taken on the requests in a timely or predefined manner.

To better support life cycle management, Tivoli Identity Manager V5.0 uses a new *recertification policy*. A recertification policy includes activities to ensure that users provide confirmation that they have a valid, ongoing need for account access.

To support recertification activities, three default reports can be generated:

- ▶ Accounts/Access Pending Recertification Report

Lists all pending recertifications for access definitions and accounts and allows filtering by account or access owner, service type, and service.

- ▶ **Recertification Change History Report**
Lists the recertification history of accounts and user accesses and allows filtering by account or access owner, recertification response, start date and end date, and other fields.
- ▶ **Recertification Policies Report**
Lists all recertification policies and allows filtering by policy target type, service type, service, access type, and access.

A recertification policy is implemented as a workflow. A default workflow is automatically built for simple policies. A recertification policy can prompt a recipient, such as a manager or system administrator, to certify periodically that users still need to use accounts. E-mail notification of the work item to be completed for recertification of accounts or accesses is generated, and “to do” activities to request the participant to take an action on the recertification to accept or reject are generated.

2.6 E-mail management design

Tivoli Identity Manager uses an external e-mail system to provide notification of events to users. Events sent through the e-mail system can be divided into two categories:

- ▶ *System notifications*, which do not require user actions. Can be disabled or enabled.
- ▶ *Manual activity notifications*, which require user actions. Cannot be disabled directly but can be disabled indirectly by setting an empty e-mail subject.

These notification e-mails can be configured to conform to a specific format, style, and content and are based on a set of configurable notification templates that are defined within Tivoli Identity Manager.

2.6.1 Notification templates

Notification templates provide a consistent notification style and content across manual and system activities, such as adding accounts, changing passwords, and doing approvals. Tivoli Identity Manager provides a standard set of templates, which can be customized as required. There is also an allowance for text and XHTML to be sent together using different MIME types and, thus, be displayed appropriately for the e-mail client that the recipient uses. These templates are all configurable through the Tivoli Identity Manager Web interface

and can provide dynamic information through standard documented tags or extended tags that are defined to a deployment.

2.6.2 Post office

The post office provides a mechanism for reducing the number of e-mail notifications a user receives regarding similar tasks in Tivoli Identity Manager. It can be configured to collect similar notifications for a period of time and combine those multiple e-mails into one notification that is then sent to a user. There is the option to enable or disable this function in Tivoli Identity Manager through the Web interface.

If the post office is enabled and the manual activities that generate notifications have **Use Group e-mail Topic** enabled, the post office intercepts notification e-mails that the system generates for those manual activities and holds them for a specified interval. When that interval expires, the post office aggregates all notifications that have the same topic, using the post office template, into one e-mail for each e-mail recipient. This aggregation reduces the volume of individual e-mails regarding notifications of the same topic that a user receives.

2.7 IBM Tivoli Identity Manager group design

A user's access rights within Tivoli Identity Manager, for example, the functions the user can perform in Tivoli Identity Manager, are governed by the roles to which the user is assigned. These roles are called Tivoli Identity Manager groups. For access control purposes these Tivoli Identity Manager groups are associated with View and Access Control Items (ACIs).

An ACI controls Tivoli Identity Manager user access by defining the access privileges of a Tivoli Identity Manager group or ACI principal. Members of a Tivoli Identity Manager group or an ACI principal (which effectively has a relationship to another entity in Tivoli Identity Manager, such as account or access entitlement) can view and perform operations on attributes within a target class (context), as defined by the scope of the ACI. The scope of an ACI is either single-level or sub-tree and apply to the branch of the Org Tree in which the ACI is placed. In Tivoli Identity Manager 5.0, there are several more succinct ACIs in place to "enable" the correct access to the correct kind of Tivoli Identity Manager user. ACIs grant or deny the ability to perform various functions.

Note: ACIs can be assigned only to Tivoli Identity Manager groups. Organizational roles cannot be assigned ACIs. ACIs grant or deny the ability to perform Tivoli Identity Manager functions. Managers, for example, can provision and manage a person in the organizational unit based on their level of access.

Tivoli Identity Manager system administrators are not controlled by ACIs because the administrator account, by default, has access to all functions in the system. All other users, by default, do not have access to any functions or features in the system. There is no default “end-user” group.

Typically, users do not need access to the Tivoli Identity Manager administrative console because Tivoli Identity Manager 5.0 provides a simplified *self-service user interface* to allow user access to certain tasks within Tivoli Identity Manager.

Within the realm of the self-service user interface a *view* affects the visibility of task panels and other elements within the self-service interface with which a Tivoli Identity Manager user typically interacts. The views can be defined for both administrator and self-service console.

After a user is logged in to a console, the user can choose from a set of tasks, depending on the Tivoli Identity Manager group they belong to and the associated views.

Views control the tasks that display; ACIs control the operations, attribute permissions, and so on that a user has when performing a task.

If a user belongs to multiple Tivoli Identity Manager groups, that user inherits the views (tasks) assigned to each group. Thus, views are cumulative and represent a union of the tasks.

By default, new Tivoli Identity Manager users who are not yet a member of a group assume a default *end user view*. This view allows them to perform self-service capabilities.

If a user belongs to multiple Tivoli Identity Manager groups, access is enabled based on the widest privilege granted to any of the Tivoli Identity Manager groups. However, if a type of ACI access is explicitly denied to a Tivoli Identity Manager group of which the person is a member, the access is denied irrespective of what was granted by other groups in that same context.

2.8 Provisioning policies design

Tracking precisely who has access to what information across an organization is a critical function of the provisioning solution. It allows control of sensitive systems, but it also should expose all accounts that have unapproved authorizations or authorizations that are no longer necessary. These inappropriate accounts pose one of the most serious threats to corporate security, because they are valid, active accounts and cannot be detected as a traditional cyber-attack if somebody uses them with malicious intent. Access rights accountability provides configuration control over all accounts and their specific authorities.

Orphan accounts are active accounts found on many systems that cannot be associated with a valid user. Improperly configured accounts are those that are associated with valid users but granted improper authorities. These accounts might appear at any time due to local administrators retaining rights to use local administrative consoles.

In enterprise-wide environments, these local consoles cannot be disabled because of their multi-operational use. The key to the control of improper and orphan accounts is on a continuous basis to associate every account with a valid user and maintain a system-of-record that details the approved authorities of the account. When a user's status within the organization changes, the user's access rights must change, too. If the account configuration changes, it must be compared with an approved configuration and policy.

The ability to control orphan accounts requires that the provisioning system link gathered account information from managed targets with authoritative information about the users. Authoritative user identity information is typically found in human resources and various databases and directories containing information about users in other businesses.

The ability to control improper accounts is much more difficult. It requires a comparison of the desired account-authority level with the reality on the managed targets. The mere existence of an account does not expose its capabilities. Accounts in sophisticated IT systems might include hundreds of parameters defining the authorities; all these are details that must be controlled.

Accounts found to be orphaned or improperly configured must be reported and corrected. Provisioning solutions should notify the proper personnel and offer to suspend, roll back, send alerts, or accept the account settings.

When developing and testing new provisioning policies, a preview function provides the ability to investigate the impact on the managed entities. This

feature can be very useful before committing a new provisioning policy to act on perhaps thousands of users and resources.

When designing provisioning policies, you have to ensure that proper access rights and capabilities for the Tivoli Identity Manager system are granted and created as follows:

- ▶ Flexible mechanisms to connect to multiple data stores containing accurate information about valid users
- ▶ Ability to load identity store information about a scheduled bulk basis
- ▶ Ability to detect and respond to identity store changes in near-real time
- ▶ Ability to retrieve account information from managed targets on a scheduled basis, both in bulk or in filtered subsets to preserve network bandwidth
- ▶ Ability to detect and report in near-real time local administrator account maintenance (creation, deletion, or changes) made directly on local resources
- ▶ Ability to compare local administrator changes against a system-of-record of account states to determine whether changes comply with approved authorities and policies
- ▶ Ability to notify designated personnel of access-rights changes made outside the provisioning solution
- ▶ Ability to compare account user IDs with valid users to identify accounts without owners (orphans)
- ▶ Ability to automatically suspend or delete a detected orphan account
- ▶ Ability to automatically suspend or roll back a reconfigured account that violates policy
- ▶ Ability to examine reports on orphan accounts
- ▶ Ability to readily view the accounts associated with a user or a resource
- ▶ Ability to assign discovered orphan accounts to a valid user

A provisioning policy confers access to many types of *managed services* (Tivoli Identity Manager, Windows, Tivoli Access Manager, Solaris, DB2, and so on) by granting a person access based on an organization (for example, a person's location in the org tree, an organizational role, or all people not in any organizational role). In other words, access to a managed target is either:

- ▶ Granted to all persons in an organization
- ▶ Granted only to persons assigned to a specified organizational role
- ▶ Granted to persons not covered by any other provisioning policies on any of the entitlement targets associated with the current policy

A provisioning policy is used to define which accounts can be created for a user and, optionally and automatically, to create the accounts on the managed target systems. A provisioning policy can also be used to define a specific approval workflow process that has to be applied to the accounts.

A *service selection policy* extends the provisioning policies by providing the ability to provision accounts based on personal attributes. In order for a service selection policy to be enforced, a provisioning policy must target it. The service selection policy then identifies the service type to target and defines provisioning based on JavaScript.

2.9 Workflow design

Tivoli Identity Manager uses three types of workflows:

- ▶ Account request workflow²
- ▶ Access request workflow
- ▶ Operation workflow

We discussed the operation workflow as part of the life cycle rule design in 2.5.5, “Life cycle management” on page 49. In this section, we focus on the account and access request workflows.

Account and access request approval and process automation is a key component in the world of rapidly and accurately changing user access rights. The approval processes are a specialized form of workflow that determine, based on organizational policy, the need to approve a requested change to access rights prior to its execution.

Many organizations today still rely on paper and e-mail forwarded in many different paths through the organization. These approaches can be very slow. Requests can sit idle in an in-box or be rejected because they are missing key information; consequently, the process must begin again.

A complete provisioning workflow solution automatically routes requests to the proper approvers and escalates to alternates if action is not taken on the request in a specified time. This workflow automation can turn a process that typically takes a week into one that takes only minutes.

For efficient automation of an approval process, in addition to escalation participants, escalation time has to be considered as well. The escalation period specifies the period within which an assigned party has to perform an activity before it is designated to a specified escalation participant.

² This workflow was known as an *entitlement workflow* in previous versions of Tivoli Identity Manager

Escalation is the period of time that the participant has to process approvals, requests for information, work orders, compliance alerts, and recertifications. If the participant does not complete the activity by the escalation date, the activity is sent to the escalation participant, and the escalation period restarts. Activity is terminated if none of the participants take action on it. An activity is sent to the system administrator only if participant resolution fails.

Some organizations also require that account or background information be added to the request as it flows through the process. This information might come from users who are involved in the process, or it might be computed or extracted from other systems.

A workflow automation tool needs to offer the following features:

- ▶ Web-based mechanism for requesting access to a system
- ▶ Automatic approval routing to the people appropriate to the system access requested and organizational structure
- ▶ Review and approval mechanisms that offer a zero-footprint client
- ▶ Ability to use predefined organizational information to dynamically determine routing of approvals
- ▶ Ability to delegate approval authority to another person
- ▶ Ability to escalate a request to an alternative approver if the allotted time elapses
- ▶ Ability for different personnel to view different levels of information based on their job duties
- ▶ Ability to request information from approval participants to define account-specific information during the process
- ▶ Ability to determine service instances where a physical account should be created
- ▶ Ability for the system to change account information in the managed resources of a specific organization
- ▶ Ability to request information from specific participants in the workflow process
- ▶ Ability to request information from external functions, applications, and data stores during the process
- ▶ Ability to easily create, design, and modify a workflow through a graphical drag-and-drop interface

A workflow is the process by which a request is approved, rejected, or sent for completion. The workflow process is defined by a workflow design. When a user places a request for a new account, new access rights, or changes to an existing

account, the request must be approved by signature authorities that are defined by a workflow design.

A workflow design can be added to an entitlement in a provisioning policy when the entitlement is defined or at a later time.

Workflow designs are built using the Tivoli Identity Manager GUI. The design created by the visual programming Java applet in the GUI produces an XML implementation under the covers.

2.10 Identity policy design

An *identity* represents the user's credentials, such as user name and password, and information about the user, including name, e-mail address, and phone number.

An *identity policy* defines how a user's ID is created. Tivoli Identity Manager generates user IDs automatically based on the identity policy. Identity policies can be set as a global policy or as a service-specific policy. If the identity policy is not a global policy, the policy can be assigned on a per-service basis (for example, it only applies to specific service types), or it can be assigned to a combination of service types or instances, or both. For example, if all user IDs must be composed of the user's first initial and last name, a global identity policy must be created for the organization. If all user IDs for a specific service must contain a certain number, a service specific identity policy must be created for the service.

2.11 Password policies design

A *password policy* sets parameters that all passwords must meet, such as length, type of characters allowed and disallowed, and so on. You can set up password policies to apply to any of the following items:

- ▶ Only one service instance or more than one service instances
- ▶ All service instances of only one service type or multiple service types
- ▶ All services, regardless of service type

Password management is the ability to control password quality and change passwords throughout an environment. As companies deploy more and more systems that contain access controls, the number of passwords required to be remembered by each user increases. This increase poses a risk to the organization as more users have a tendency to write down their passwords in

order to keep track of them. A costly side effect of this is the increased workload on the help desk to reset forgotten passwords.

Note: Research shows that approximately 30% of total calls to the average help desk are for password-reset assistance.

Password strength is another challenge for many organizations. Malicious hackers possess effective tools and techniques for cracking poorly constructed passwords.

Password strength can be defined by password rules to which a password must conform. For example, password strength rules might specify that the minimum number of characters of a password must be five and the maximum number of characters must be ten.

Additionally, the password policy might specify that an entry is disallowed if it appears in a dictionary of unwanted terms. To select this choice in the user interface, you must first load a dictionary.ldif file into Tivoli Identity Manager.

You can specify the following standard and other rules for passwords:

- ▶ Minimum and maximum length
- ▶ Character restrictions
- ▶ Frequency of password reuse
- ▶ Disallowed user names or user IDs

Organizations desire to enforce stronger password formation rules across the enterprise, but must balance that desire against poor user experience and increases in forgotten passwords.

If default password rules cannot fulfill requirements, Tivoli Identity Manager provides an option to create custom Java classes by implementing the `com.ibm.passwordrules.Rule` interface.

Password management capabilities can also enable users to self-service their accounts. Users visit a Web-based system, authorize themselves, and then can reset or synchronize passwords on all of their accounts. Further, the passwords they select can be evaluated against rules on their formation to ensure uniform conformance with organizational password policies. A user typically has multiple accounts and passwords. The ability to synchronize passwords across platforms and applications provides ease of use for the user. It can also improve the security of the environment, because each user does not have to remember multiple passwords and is, therefore, less likely to write them down.

Key points to password management include:

- ▶ Providing user self-service through the Web without logging in to the identity management application
- ▶ Challenge-response system to authenticate a user with a forgotten password by using shared secrets
- ▶ Ability to implement password formation rules to enforce password strength uniformly across the organization
- ▶ Ability to synchronize passwords for multiple systems to the same value to reduce the number of different passwords to be remembered by the user
- ▶ Delivery of password-change status (success or failure) to requestor
- ▶ Ability to securely deliver passwords to users for new accounts

2.12 Security model design

Security model design in Tivoli Identity Manager is the way to evaluate and enforce business processes and rules for granting access. There are several commonly-found access control models in a centralized identity management solution. Tivoli Identity Manager can be used to support these different types of access control models:

- ▶ **Role Based Access Control (RBAC)**
Grants access privileges to users based on the work that they do within an organization. The model allows an administrator to assign a user to a single or multiple roles according to their work assignments. Each role enables access to specific resources.
- ▶ **Discretionary Access Control (DAC)**
Enables the owner of a resource to decide whether to allow a specific person access to the owned resource. This system is common in distributed environments that have evolved from smaller operations into larger ones.
- ▶ **Mandatory Access Control (MAC)**
Enables grouping or marking resources according to a sensitivity model. This model is most commonly found in military or government environments. One example would be the markings of Unclassified, Restricted, Confidential, Secret, and Top Secret. The privileges that a user has to view certain resources depend on the clearance level of the user.

The model that an organization uses depends on factors such as externally mandated policies, maturity of existing identity management processes, range of

identity management target systems, future requirements, number of users managed, and risk assessment and return on investment statistics.

2.12.1 Access provisioning models

Depending on the business needs of an organization, Tivoli Identity Manager provides the following *access provisioning models*:

- ▶ Role-based

Account entitlements to managed services are provisioned automatically based on the user's roles in the organization. To some degree, the role-based provisioning can be used to support a role-based access control model when access control is not centrally managed by a common access control system. The automation between the role and accounts and groups on the target resource, and strict enforcement role relationship, ensures that access to the IT resource is based on the user's role.

- ▶ Request-based

In request-based provisioning, users and their managers search for and request access to specific applications, privilege levels, or resources with a system. The requests are validated by workflow-driven approvals and audited for reporting and compliance purposes. Unlike role-based provisioning, request-based provisioning is based on concept of access entitlements that are direct links to assets.

Request-based provisioning is often used to support DAC and MAC access control with a combination of appropriate approval processes. Sometimes there might be mixed usage of the two models for different sets of users in the organization, or for different sets of target services.

- ▶ A hybrid provisioning model

The hybrid model of provisioning resources combines request and role-based approaches, which are both supported by Tivoli Identity Manager.

For a subset of employees or managed systems, an organization might want to automate access with role-based assignment, and also handle all other access requests or exceptions through a request-based model. Some organizations might start with manual assignment, and evolve toward a hybrid model, with an intention of a fully role-based deployment at a future time.

Other organizations might find it impractical for business reasons to achieve complete role-based provisioning, and target a hybrid approach as a desired goal. Still others might be satisfied with only request-based provisioning, and not want to invest additional effort to define and manage role-based, automated provisioning policies.

2.12.2 Role-based access control

Role-based access control (RBAC) is a method of enforcing business processes and rules to grant access rights to users based on their assignment to a defined role in the organization. Provisioning solutions that embody RBAC or other types of rules that assign access rights to users based on certain conditions and user characteristics are examples of user administration policy automation.

Automation is key to managing large numbers of users across disparate resources and assigning, monitoring, and revoking user entitlements. The solution should enable users to be defined as members of groups, including roles. Entitlements to resources for these groups of users are defined in the security policies. Any change to information about a user should be evaluated to determine whether it alters the user's membership to a group. If there is a change in the user's information, policies must be reviewed and changes to entitlements must be put into place immediately. Likewise, a change in the definition of the set of resources in a policy might also trigger a change in entitlements.

The following abilities should be included in user administration policy automation:

- ▶ Associate access-rights definition with a role within the organization.
- ▶ Assign users to one or more Tivoli Identity Manager groups.
- ▶ Implicitly define subsets of access to be unavailable to a role.
- ▶ Explicitly assign users individual access rights.
- ▶ Dynamically and automatically change access rights based on changes in user roles.
- ▶ Define implicit access rights available to users in a role upon their request and approval.
- ▶ Use defined organizational information to dynamically determine routing of approvals.
- ▶ Detect, evaluate, and respond to user authority changes made directly to a managed resource.
- ▶ Report on roles, rights associated with roles, and users associated with roles.
- ▶ Set designated times for changes in access rights or policies.
- ▶ Create unique user IDs consistent with policies and not in current use or previous use by the organization.
- ▶ Create user authorizations extending an existing account.
- ▶ Support for mandatory and optional entitlements (optional entitlements are not automatically provisioned but can be requested by a user in the group).

- ▶ Support for entitlement defaults and constraints (each characteristic of an entitlement can be set to a default value, or its range can be constrained, depending on the capabilities of the entitlement to be granted).
- ▶ Create a single account with multiple authorities governed by different policies.
- ▶ Create user IDs using a set of consistent algorithms defined by the organization.

2.13 Customization design

Customization requires a certain action on the product that can change the basic behavior of the product. Usually, the customization tasks for Tivoli Identity Manager project include:

- ▶ Human resources data feed
- ▶ Graphical user interface (GUI) customization
- ▶ Workflows
- ▶ Reports customization
- ▶ Self-registration
- ▶ Custom adapter design and implementation
- ▶ E-mail customization

In this section, we concentrate on the GUI customization. We cover the other topics separately in the book. Understanding GUI customization helps to drive the project plan and design of the Identity Management System.

2.13.1 Graphical user interface

The Tivoli Identity Manager Server Java application comes with two type of user interfaces:

- ▶ Administrative console

The administrative console provides an advanced set of administrative tasks, such as managing roles, policies, reports, and so on. The interface also features multitasking capabilities.

- ▶ Self-service (self-care)

The self-service user interface focuses on users and provides simplified user interface for self-service functions such as update of personal information and passwords, view requests, complete and delegate activities, and request and manage their own accounts and accesses.

Both interfaces are customizable to satisfy its own purpose and customer needs, and to be able fits into customer environment.

Administrative console

The Tivoli Identity Manager administrative console user interface is customizable, allowing organizations to integrate a common corporate appearance while maintaining the flexibility to perform administrative identity tasks integral to their roles and responsibilities.

The administrative console provides sets of tasks, each tailored for the needs of the default administrative user types:

- ▶ System administrator
- ▶ Service owner
- ▶ Help desk assistant
- ▶ Auditor
- ▶ Manager

System administrators can easily customize which tasks the different types of users can perform. To control user access to accounts and tasks, for example, use a default set of *user groups*, *access control items*, and *views*. You can also customize user access by defining additional user groups, views, and access control items. For more information, see Chapter 4, “Implementation” on page 101.

Some of the features can be customized through properties files and replaceable image files. The properties files that allow you to define the appearance of the Tivoli Identity Manager administrative console user interface are:

- | | |
|-------------------------------|---|
| ui.properties | Controls the appearance of the header, footer, and home page, and configures the title, number of pages displayed, and the number of search results returned. |
| helpmapping.properties | Controls the redirection and mapping of administrative console HTML help. |

You can change the appearance of the administrative console user interface by customizing the following items:

- ▶ Banner content
- ▶ Footer content
- ▶ Administrative console home page
- ▶ Title bar
- ▶ Redirecting help content
- ▶ The number of items displayed on panels

Forms

A part of the administrative console is the form designer, it can help customize different forms. Only individuals who are part of the administrator group can access this feature.

Each object in Tivoli Identity Manager has forms associated with it. Tivoli Identity Manager provides default forms to create, view, and modify system entities. The form designer allows system administrators to manage all entity forms from one location.

System administrators can customize forms for the following system entities using the form designer:

- ▶ Account
- ▶ Admin Domain
- ▶ Business Partner Organization
- ▶ Business Partner Persons
- ▶ Tivoli Identity Manager User
- ▶ Location
- ▶ Organization
- ▶ Organizational Unit
- ▶ Person
- ▶ Service

Each form category folder has associated object profiles that represent system entities. Each object profile is associated with a form template.

Default form templates are generated from an entity's configuration. Form templates have at least one tab and one form element. A tab is a container for grouping form elements. A form element is a system entity attribute. Each tab consists of a label describing the group and at least one form element. Each form element consists of a label describing its data and the data input format. Form elements are listed in the order the elements are presented on the form. After the form is customized as the final design consider updating the property key and value pairs in the CustomLabels.properties file that are used by the Tivoli Identity Manager GUI to display the label text for forms.

To learn more about the form designer and how to use it, consult the Tivoli Identity Manager Information Center:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

Self-service user interface

The Tivoli Identity Manager self-service user interface is highly customizable, allowing organizations to integrate a common corporate appearance while maintaining the flexibility to perform self-care identity management tasks integral to their roles and responsibilities.

You can define and customize the self-service interface in two ways, by using the built-in console framework and by directly modifying files installed within Tivoli Identity Manager:

- ▶ Built-in console features:
 - Access control items (ACIs)
 - Views
- ▶ Modifiable files:
 - Properties files
 - Cascading style sheet (CSS) files
 - A subset of Java server pages (JSP™) files
 - Image files

You need to backup any modifiable files for recovery purposes before making customization changes to Tivoli Identity Manager.

Built-in console features

From the self-care home page, the following task panels are available, depending on the authority the system administrator has granted through the administrative console:

Action Needed	A list of tasks that require completion.
My Password	A list of tasks to change passwords. If password synchronization is enabled, users can enter one password that is synchronized for all of their accounts. A user can reset a forgotten password by successfully responding to forgotten password questions, if forgotten password information is configured in the system.
My Access	A list of tasks to request and manage access to folders, applications, roles, and other resources.
My Profile	A list of tasks to view or update personal information.
My Requests	A list of tasks to view requests that a user has submitted.
My Activities	A list of activities that require user action. Users can also delegate activities.

Modifiable files

All self-service properties configuration files that are used to define behavior and customization of self-service interfaces are located in the ITIM_HOME\data path. The following properties files are included:

- ▶ **SelfServiceUI.properties**

Controls the layout of the user interface (banner, footer, navigation bar, tool bar), the number of pages displayed, and the number of search results that are returned.

It also, allows you to configure the items that are available in the Search By box for a user search in the self-service interface.

The file allows you to enable direct access to the Expired Password change screen and bypass the self-service login page under certain conditions. The property key that allows this is `ui.directExpiredChangePasswordEnabled`.

- ▶ **SelfServiceScreenText.properties**

Provides the text that is displayed on the self-service user interface.

- ▶ **SelfServiceScreenText_ *language*.properties**

Provides the language-specific text that is displayed on the self-service user interface. By default this file is `SelfServiceScreenText_en.properties` which contains the English language bundle.

- ▶ **SelfServiceHomePage.properties**

Defines the sections of the self-service user interface home page and the order in which they will be displayed.

- ▶ **SelfServiceHelp.properties**

Defines the links to HTML help pages that appear on the self-service user interface. The HTML files are located in the following directory:

`WAS_PROFILE_HOME\installedApps\node_name\ITIM.ear\itim_self_service_help.war`

You can redirect help by modifying the information in this file.

- ▶ **SelfServiceScreenTextKeys.properties**

Provides label keys that are displayed on the self-service user interface. This file can be used to assist with customization of screen text by providing a template to develop labels and instructions.

Web interface layout

You can completely change the layout in the self-service user interface using customization. You can enable and disable high-level layout elements from the display in the self-service user interface using settings in the `SelfServiceUI.properties` file. The complete layout contains the following elements:

- ▶ Banner
- ▶ Toolbar
- ▶ Footer
- ▶ Navigation
- ▶ Content

Turning on and off page elements can give a variety of layout options. The only required page element is the content element, which contains the tasks and task panels.

To show or hide a page element, change the `ui.layout.showname` property in the `SelfServiceUI.properties` file. For example, `ui.layout.showBanner` controls the display of the banner section. Setting a property to `true` indicates the element is included in the page, and setting a property to `false` indicates that the element is not included in the page.

In addition, you can change the appearance of the self-service user interface by customizing the banner, footer, tool bar, and navigation bar, which are JSP fragments that are included in the layout of the Web page when the JSP is rendered.

Table 2-1 displays a list of layout elements and their corresponding files. You can find these files in the following directory:

WAS_PROFILE_HOME\installedApps\node_name\ITIM.ear\itim_self_service.war\custom

Table 2-1 *Layout elements and file names*

Layout element	File name
Banner	banner.jsp
Footer	footer.jsp
Navigation bar	nav.jsp
Tool bar	toolbar.jsp

Home page

You can also change the home page in the self-service user interface using customization. The home page refers to the main page that is loaded in the content layout element after a user logs in to the self-service user interface.

The home page has a JSP fragment that is included in the layout of the Web page. This layout information is stored in the Home.jsp file in the WAS_PROFILE_HOME\installedApps\node_name\ITIM.ear\itim_self_service.war\custom directory. Some context of the home.jsp file can be altered by modifying SelfServiceHomePage.properties.

Customizing style sheets

The other method to change the appearance of the self-service user interface is by customizing *cascading style sheets*. A cascading style sheet (CSS) is used to style the appearance of the self-service user interface. You can edit the style sheet to modify the fonts, colors, and other styles that are associated with the self-service user interface.

The default deployed CSS files are compressed and optimized with bandwidth in mind for scalability. The non-optimized versions (with whitespace/formatting intact) can be found in the ITIM_HOME\defaults\custom directory. The CSS files stored in the following directory are unsuitable for editing:

WAS_PROFILE_HOME\installedApps\node_name\ITIM.ear\itim_self_service.war\custom

You need to copy the default files stored in the ITIM_HOME\defaults\custom directory to another directory, edit the style sheets, and then copy your changed files to the following directory:

WAS_PROFILE_HOME\installedApps\node_name\ITIM.ear\itim_self_service.war\custom

2.14 System architecture

An *architecture* describes the structure of the system and the relationships between components. Usually, the system architecture is examined from two different points:

Logical architecture The *component*, or logical, structure of the architecture. This describes the components (both product and external that interface to the product) and the relationships between the products from a data-exchange perspective. This is sometimes referred to as the *functional* architecture.

Physical architecture The *physical structure* of the architecture. This describes the product placement, the hardware requirements, the network-level design (including communications protocols, firewall placement, and port use), and other infrastructure components (such as database placement and middleware use) involved in the operational deployment of the system. This is sometimes referred to as the *operational* architecture.

You need to consider both the logical and physical architecture when planning the Tivoli Identity Manager installation. You also need to create a *system architecture document*. This document needs to describe the interaction between components, the data flow, and the overall security. The security of the Tivoli Identity Manager system consists of various aspects that are described in the ISO/IEC 27002 standard that is entitled *Information technology - Security techniques - Code of practice for information security management*.³

The standard contains the following main sections, and each section discusses information security controls and their objectives:

1. Risk assessment: Security risk analyzes and assessments
2. Security policy: Company and management direction
3. Organization of information security: Governance of information security
4. Asset management: Classification of information assets and their inventory
5. Human resources security: The security aspects for HR. Handling employees information and aspects of joining, moving and leaving an organization
6. Physical and environmental security: Protection of the physical resources: buildings, hardware, physical access to the restricted area and so on

³ The current standard is a revision of the version first published by ISO/IEC in 2000 known as ISO 17799, which was a word-for-word copy of the British Standard (BS) 7799-1:1999.

7. Communications and operations management: The management of technical security controls in systems and networks
8. Access control: Restriction of access rights to networks, systems, applications, functions and data
9. Information systems acquisition, development, and maintenance
10. Information security incident management: Anticipating and responding appropriately to information security breaches
11. Business continuity management: Protecting, maintaining and recovering business-critical processes and systems
12. Compliance: Ensuring conformance with information security policies, standards, laws and regulations

2.14.1 High availability

High availability is a feature that allows an overall system to be available even if some of its components fail. To create a system that is highly available, you have to duplicate hardware and software components of the system. You also need a logic to recognize the failure of an individual component and to overcome the problem. In addition, high-availability systems need to have the capability to reset a system to normal functionality after a failed component is repaired or replaced with a new one.

Tivoli Identity Manager Server basically is a Java application that runs on IBM WebSphere Application Server. If you configure your WebSphere Application Server cluster to be highly available, your Tivoli Identity Manager application benefits automatically and becomes highly available as well.

In case of the overall Tivoli Identity Manager system high availability, every component needs to be highly available. Figure 3-1 on page 80 depicts the typical components in a Tivoli Identity Manager environment, which include database and directory servers.

You need to pay special attention to the design of the directory server high availability solution. Usual practice is to place a load balancer in front of a directory server cluster. However, the LDAP traffic should be routed 100% only to one node and fail over to a second node must only occur if the first node is no longer available (so this solution falls more into a failover scenario). Also, after the node recovers, you need to fall back to the first node manually after the directory server is fully synchronized. This design ensures that directory server data stays synchronized due to race conditions when Tivoli Identity Manager writes heavily to the directory server.

To enable the Tivoli Identity Manager environment for high availability, you need to include the supported directory servers, databases, and IBM WebSphere Application Server. For a further discussion about high availability, see *Identity Management Advanced Design for IBM Tivoli Identity Manager*, SG24-7242.

2.14.2 Archival and backup

High availability, however, cannot replace a system backup and recovery strategy or archival procedures. In this section, we discuss the critical software and data components that are required by the Tivoli Identity Manager application. The archival and backup of data on the managed resources that Tivoli Identity Manager manages (for example, Windows Active Directory®) is different for each type of managed resource and is not within the scope of this discussion.

Archival

Tivoli Identity Manager stores various types of data that is required for its operations. Archival of this data might be required at various stages for varying business and operational needs. Examples of such needs include increasing system performance and efficient usage of available system resources. As a result, archival of data in this context usually results in the deletion of data from the relevant live Tivoli Identity Manager data stores.

Archival of live Tivoli Identity Manager data that is required by the application is not practical in most cases. Archiving live data can be done, but it is not common practice. (Backing up live data, however, is common, as we discuss in the next section.) Commonly, requirements exist to have an archival strategy for Tivoli Identity Manager data that is of no use for day-to-day operational and functional requirements but that must be retained for other reasons, such as audit and compliance. For more details about archival, see *Identity Management Design Guide using IBM Tivoli Identity Manager*, SG24-6996.

Backup

A good backup strategy allows for a system to be restored to a known system state for a particular instance in time. In the case of Tivoli Identity Manager, this backup strategy includes the data within the Tivoli Identity Manager data stores and the relevant file systems on which all Tivoli Identity Manager software components are installed.

File system

You need to ensure that the file systems of the following components are backed up appropriately, as per the procedures of the environment in question:

- ▶ Operating system-specific files of each system that has a Tivoli Identity Manager software component installed
- ▶ Each Tivoli Identity Manager application instance
- ▶ Each application server hosting a Tivoli Identity Manager application instance
- ▶ Each LDAP instance, not including the LDAP data itself
- ▶ Each relational database instance, not including the data that is stored within the databases
- ▶ Each Tivoli Identity Manager adapter that interacts with a managed resource
- ▶ Each Tivoli Identity Manager reverse password synchronization component
- ▶ Any Tivoli Directory Integrator components that interact with Tivoli Identity Manager

IBM Tivoli Identity Manager application data

You need to ensure that the data stored within the following components is backed up appropriately, per the data backup procedures of the component in question:

- ▶ Tivoli Identity Manager LDAP
- ▶ Tivoli Identity Manager relational database

Also, during the backup process, be sure to consider any migration to a new version of the product, as discussed in Chapter 8, “Maintenance” on page 205.

2.15 Adapter project plan

Adapters communicate with the Tivoli Identity Manager Server and are responsible for translating add, change, delete, and other requests into actions on the managed targets. Tivoli Identity Manager comes with a set of common adapters for various managed targets, which include IBM Tivoli technical support. Some of the predefined adapters from previous versions of Tivoli Identity Manager are now obsolete (for example, the Windows NT® adapter because Microsoft has also withdrawn their support for Windows NT). IBM Tivoli no longer provides support for these obsolete adapters. So, existing adapter support might be something that you want to consider in your adapter project plans.

In some cases, you might want to develop custom adapters. You can use *custom adapters* to manage accounts on platforms for which no predefined adapter is available. The preferred method for developing a custom adapter is to use Tivoli Directory Integrator. Tivoli Directory Integrator provides connector components for many platforms, and additional connectors can be developed using Java or JavaScript. For more details about custom adapter design, see the product documentation and examples that come with Tivoli Identity Manager.

We cover basic adapter installation and configuration steps in Chapter 3, “Installation” on page 79. As a part of the adapter installation, you also need to install (import) a service profile into Tivoli Identity Manager Server for each managed target. The service profile contains various attributes, depending on the adapter that it is controlling. Some of these attributes can be mandatory.

In a case where the Tivoli Identity Manager implementation requires a large number of various adapters, consider preparing a plan for tracking the adapter installation by platform.

When you are planning the adapter deployment, there is no simple guideline about the hardware requirements of the managed target. The basic adapter installation does not consume too much disk space, but if you want to use the event notification feature, the adapter can maintain a local account database that increases the necessary disk space. Also, a large number of accounts can have an impact on network traffic, CPU, and memory requirements, especially during large data transfers such as during a reconciliation process.

2.16 IBM Tivoli Identity Manager project planning

As you can see, a Tivoli Identity Manager solution can be complex and might require many different activities, from installation and adapter deployment to customization, workflow design, and so on. The best approach to deploy a Tivoli Identity Manager solution within an enterprise is through logical stages that add value incrementally.

This section describes an incremental approach to delivering a Tivoli Identity Manager management solution. It decomposes the overall implementation stage into distinct phases and defines the scope, outcomes, and benefits of each phase. The rationale behind this delivery strategy is to restrict each phase to manageable chunks, while still delivering tangible business value. This strategy

enables the enterprise to reach key milestones as early as possible and also to become self-reliant as soon as possible. We begin by taking a closer look at the Identity Management delivery strategy, as shown in Figure 2-2.

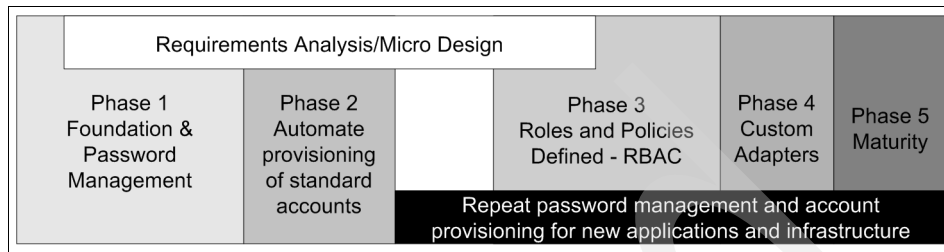


Figure 2-2 Identity Management delivery strategy

Typically, phase 1 and 2 cover infrastructure accounts. These accounts are the accounts in which the majority of the user base is represented (such as Windows, UNIX, and IBM z/OS® RACF® accounts). After the initial iteration of phase 1 and 2 is complete, you can combine and repeat these phases to bring other standard systems under centralized identity management. This can be a parallel activity, so over a relatively short period of time, all standard systems come under centralized control.

Phase 3 is dependant on the business definition of roles and policies. It focuses on business unit-based and role-based provisioning. Lessons learned enable refinement and replication across other business units. This phase is the most complex and far reaching and, ultimately, has the highest long-term rewards.

Phase 4 brings in those systems that require a custom agent and completes the solution with all agreed systems and applications under centralized identity management.

Phase 5 marks the total integration of centralized identity management into the enterprise's business processes and IT environment.

After the foundation phase 1 is complete, it is possible, if business needs require, to change the focus and content and to commence a project phase to deliver any element of the subsequent phases. Certain prerequisites might be required, but none of these prerequisites will disallow the activity, although they might extend overall implementation time and costs.

Phase 1: Foundation and password management

The first phase focuses on delivering high-visibility, high-value benefits quickly and with minimal impact on existing systems. These benefits are typically realized by implementing self-service password reset and by managing orphaned accounts.

The scope of phase 1 includes:

- ▶ Password management
- ▶ Supported applications and systems
- ▶ Baseline reporting
- ▶ Large or small user target coverage
- ▶ HR feed established
- ▶ Orphan account control
- ▶ Single action to close or suspend user accounts

The expected outcomes of phase 1 are:

- ▶ Password management using synchronization, reset, and self-service across managed platforms
- ▶ Organizational tree established
- ▶ Eliminate risks from backdoor access
- ▶ Necessary reporting available

The benefits delivered by phase 1 include:

- ▶ High visibility of the solution
- ▶ Large benefits gained among the users and in the central user administration and support desk
- ▶ Compact delivery time

Phase 2: Automatic provisioning for infrastructure accounts

The second phase automates the provisioning process for *standard* accounts and resources.

The scope of phase 2 includes:

- ▶ Automatic provisioning of standard accounts and workflow
- ▶ Consistent GUI for administration
- ▶ Consistent account creation
- ▶ Full audit trail
- ▶ Simple workflow introduced
- ▶ Foundations for role-based access control (RBAC)

The expected outcomes of phase 2 are:

- ▶ User registration automatically updated
- ▶ Reduced administration
- ▶ Necessary reporting for external parties available
- ▶ Consolidation of users
- ▶ Organizational structure
- ▶ HR feed creating new users

The benefits delivered by phase 2 include:

- ▶ High visibility of the solution
- ▶ Large benefits gained among the end users and in the central user administration
- ▶ Improved security and potentially reduced user-based license costs

Phase 3: RBAC for services and applications as delivered

The third phase focuses on implementing role-based access control. By this time, the analysis of business requirements is complete, the business roles are mapped to access rights, and security policies are defined. These roles, rights, and policies are used to deliver automatic role-driven provisioning and deprovisioning of access rights.

The scope of phase 3 includes:

- ▶ Role-based account management
- ▶ Rule set for automated creation and deletion of user accounts
- ▶ Rule set for organizational changes
- ▶ Full workflow for account management
- ▶ Focused on small community

The expected outcomes of phase 3 are:

- ▶ HR feed for managing user accounts, high demands on data quality.
- ▶ Organizational chart might need refining.
- ▶ Administration by role management introduced.
- ▶ Requires input and buy-in from application/system owners.

The benefits delivered by phase 3 include:

- ▶ Time-consuming tasks replaced by automation
- ▶ Large benefits gained by the application owners
- ▶ Delegated administration possible
- ▶ Improved control from detailed reporting

Phase 4: Develop custom adapters

The fourth phase continues the work of phases 2 and 3 with nonstandard applications account types that require custom integration.

The scope of phase 4 includes:

- ▶ Custom adapters and extensions.
- ▶ Custom developed adapter.
- ▶ Start program to extend RBAC to cover all companies.
- ▶ Eliminate all administration of user accounts outside of the identity management solution.
- ▶ Workflow supports authorization management.

The expected outcomes of phase 4 are:

- ▶ Templates for later roll-out established
- ▶ All significant applications covered

The benefits delivered by phase 4 include:

- ▶ One interface for *all* user administration
- ▶ Scheduled re-organizations with shorter non-productive time for the end user
- ▶ Fast activation and deactivation of users
- ▶ Time-consuming tasks replaced by automation

Phase 5: Maturity

By now, the initial deployment of the identity management solution is complete. The fifth phase covers the ongoing evolution of the identity management solution as new business roles, applications, and infrastructure are added to the IT environment.

The scope of phase 5 includes:

- ▶ The enterprise can repeat new instances of adapter installations and can integrate into appropriate policies.
- ▶ The enterprise can self-maintain the solution to reflect changing business demands.

The expected outcomes of phase 5 are:

- ▶ Role-based access control fully enabled
- ▶ Only *run-out* applications excluded, if any

At the completion of this phase, the organization can expect to realize the full potential of an identity management solution, such as:

- ▶ Easing compliance with security audits
- ▶ Consolidating control of the user management processes
- ▶ Eliminating inconsistencies from human error and *management by mood*⁴

- ▶ Reducing training costs and education requirements
- ▶ Reducing help desk and overall administration costs
- ▶ Involving less people in day-to-day management
- ▶ Dividing work along organizational/departmental structures
- ▶ Improving response to user changes
- ▶ Taking advantage of user information in all business processes

If you have to migrate a Tivoli Identity Manager environment to a new version, take into account an approach similar to this overall project plan. Analyze the current system needs, and check and document migration requirements. We provide more detail about migration requirements in Chapter 8, “Maintenance” on page 205.

⁴ Management by mood refers to personal administrative favors or quick-and-dirty solutions that do not comply with any policy, for example, an administrator grants you root privileges just for one week because the administrator knows you personally and trusts you.

Installation

In this chapter, we provide an overview of the installation process and requirements for middleware software. We also cover the installation process for adapters and for communication between adapters and IBM Tivoli Identity Manager Server. This chapter addresses high-level installation issues. For more detailed, step-by-step installation instructions, refer to *IBM Tivoli Identity Manager Server Version 5.0 Installation and Configuration Guide, SC32-1562*.

3.1 IBM Tivoli Identity Manager components overview

Figure 3-1 depicts the basic logical components of which a Tivoli Identity Manager installation consists.

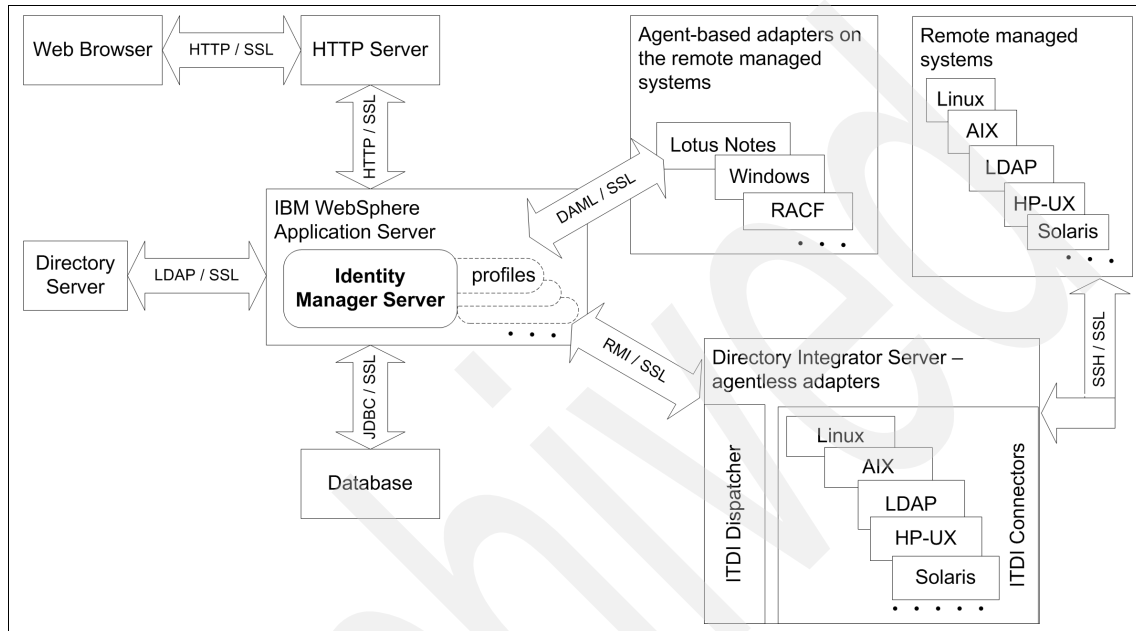


Figure 3-1 Tivoli Identity Manager logical components overview

The Tivoli Identity Manager Server application runs on IBM WebSphere Application Server and communicates with adapters on remote systems.¹ The Tivoli Identity Manager application runs on a single-server configuration with the WebSphere Application Server base product. However, Tivoli Identity Manager can also run in a larger cluster configuration that is composed of one or more WebSphere Application Servers and of a deployment manager that manages a cluster.

Tivoli Identity Manager stores transactional and historical data in a database server. For example, the Tivoli Identity Manager provisioning processes use a relational database to maintain their current state and their history. A type 4 Java Database Connectivity driver (JDBC™ driver) connects the Tivoli Identity Manager Server to a database. The DB2 and Microsoft SQL type 4 JDBC drivers are bundled with the Tivoli Identity Manager installation program. For an Oracle

¹ In the earlier versions, the adapter was known by the names *agent* and *connector*.

database, you must obtain this JDBC driver (ojdbc14.jar) from your Oracle Database Server installation.

Along with using a relational database, Tivoli Identity Manager stores the current state of the managed identities in an LDAP directory, including user account and organizational data. It is always recommended that Tivoli Identity Manager has its own database and LDAP server, due to the high volume of data exchange between these two components and Tivoli Identity Manager Server.

Finally, an HTTP server, such as IBM HTTP Server, and an IBM WebSphere Web server plug-in enable browser-based access to the Tivoli Identity Manager Server.

Figure 3-1 illustrates that Tivoli Identity Manager supports the use of two type of adapters:

- ▶ *Agent-based* adapters, which must reside on the managed resource to administer accounts. Communication between adapter and Tivoli Identity Manager Server is usually through DAML protocol. So, these adapters are often called *DAML-based adapters*.
- ▶ *Agentless* adapters can reside on a remote server to administer accounts. For example, the UNIX/Linux adapter is an agentless adapter.

IBM Tivoli Directory Integrator is an optional installation component that is used for housing of agentless, RMI-based (Remote Method Invocation) adapters. The tool is also used for complex HR feeds (a load of person data into Tivoli Identity Manager) from typical resources or from multiple resources. Tivoli Directory Integrator can be installed on a separate server (usually called *Adapter server*), or it can be co-located on the same server that runs WebSphere Application Server and Tivoli Identity Manager Server.

The RMI Dispatcher is a Tivoli Directory Integrator component that enables the Tivoli Identity Manager Server to communicate with a Tivoli Directory Integrator-based adapter using RMI. The RMI Dispatcher is the request handler inside Tivoli Directory Integrator for the Tivoli Directory Integrator-based adapters. The RMI Dispatcher is not installed with the base Tivoli Directory Integrator product and must be installed separately in order for the Tivoli Directory Integrator-based adapters to run.

Supported software and operating systems

Tivoli Identity Manager (with the supported fix packs or service packs as listed in the *IBM Tivoli Identity Manager Version 5.0: Product overview*,² is supported on the following operating systems:

- ▶ IBM AIX® 5L™ V5.3
- ▶ Sun Solaris 10
- ▶ Microsoft Windows 2003 Server Enterprise, Standard Editions
- ▶ Red Hat Enterprise Linux 4.0 and 5.0 Intel®, System p® and System z®
- ▶ SUSE® Linux Enterprise Server 9.0 and 10.0 Intel, System p and System z
- ▶ HP-UX 11i v2

Tivoli Identity Manager adapters for Version 5.0 support a wide range of OS platforms, databases, and applications, including OS/400®, RACF, DB2, Sybase, Oracle, IBM Tivoli Access Manager, Lotus® Domino® Server, and Microsoft Active Directory.

Tivoli Identity Manager requires Java Runtime Environment Version 1.5, which is bundled with WebSphere Application Server Version 6.1 in the WAS_HOME/java directory. This is the only supported Java (according to the IBM Tivoli Identity Manager Information Center) because the JRE™ requirements for using a browser to create a client connection to the Tivoli Identity Manager Server are different than the JRE requirements for running the WebSphere Application Server.³

Tivoli Identity Manager also requires the following minimal versions of middleware software:

- ▶ WebSphere Application Server V6.1 Fix Pack 9, and the included IBM HTTP Server
- ▶ Databases:
 - DB2 Universal Database™ (UDB) Enterprise Edition V9.1 Fix Pack 2⁴
 - Oracle 10g release 2
 - Microsoft SQL 2005 Enterprise Edition⁵

² The Product Overview PDF file can be obtained at:
http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/pdf/im50_oview.pdf

³ The IBM Tivoli Identity Manager Information Center is available at:
http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/cpt/cpt_ic_release_oview_jre.htm

⁴ Exception is HP-UX 11i v3 (PA-RISC, Itanium®) that requires Fix Pack 3.

⁵ Supported only on Microsoft Windows 2003 Server.

- ▶ Directory servers (LDAP servers)
 - IBM Tivoli Directory Server V6.0 fix 5 or 6.1
 - Sun Java™ System Directory Server 5.2.x (formerly Sun ONE Directory Server)
- ▶ IBM Tivoli Directory Integrator Version 6.1.1, Fix Pack FP0003 or higher
- ▶ Browsers:
 - Microsoft Internet Explorer® 7.0 and 6.0 Fix Pack 1
 - Mozilla 1.7.13
 - Firefox, Version 1.5.0.7 and 2.0

Note: For the latest information about specific levels and required fix packs for OS and middleware components, always refer to *IBM Tivoli Identity Manager Version 5.0: Product overview*, Release Notes section.

3.2 SSL communication overview

When you deploy Tivoli Identity Manager into a production environment, we strongly recommend that all communication between Tivoli Identity Manager components, depicted in Figure 3-1 on page 80, is secured. The *Secure Sockets Layer* (SSL) protocol is used to secure communication in the Tivoli Identity Manager environment, because it provides endpoint authentication and communications privacy over the network using cryptography.

In a typical use case where a browser connects to a Web server application, only the server is authenticated while the client remains unauthenticated. In this case, the server provides a set of symmetric keys for encryption that are only valid while the particular session between the browser and the server is active.

Mutually authenticated SSL sessions, as used between the Tivoli Identity Manager components, require the use of public key cryptography between the participants. Encryption of data is provided using the public-key cryptography algorithm, also known as *asymmetric key cryptography*.

Public key cryptography allows users to communicate securely using a pair of cryptographic keys, designated as *public key* and *private key*, which are related mathematically.⁶ The private key is kept secret, whereas the public key can be widely distributed. A sender encrypts a message using the public key of the recipient who can decrypt the message using the private key. Thus, participants

⁶ For more details about public key cryptography refer to:
http://en.wikipedia.org/wiki/Public_key

in a mutually authenticated SSL communication have to exchange the public key before they can encrypt any information.

Also, public-key cryptography is built on trust. The recipient of a public key needs to have confidence that the key really belongs to the sender and not to a fraudulent party. *Digital certificates* provide that confidence. A digital certificate serves two purposes:

- ▶ it establishes the owner's identity.
- ▶ It makes the owner's public key available.

A digital certificate is issued by a trusted authority—a *certificate authority* (CA)—and is issued only for a limited time. When its expiration date passes, the digital certificate must be replaced.

The digital certificate contains specific pieces of information about the identity of the certificate owner and about the certificate authority, including:

- ▶ The owner's distinguished name
- ▶ The owner's public key
- ▶ The date the digital certificate was issued
- ▶ The date the digital certificate expires
- ▶ The issuer's distinguished name
- ▶ The issuer's digital signature

You can use self-signed certificates to test an SSL configuration before you create and install a signed certificate issued by a certificate authority. A self-signed certificate contains a public key, information about the owner of the certificate, and the owner's signature. It has an associated private key, but it does not verify the origin of the certificate through a third-party certificate authority.

A self-signed certificate serves as both a CA certificate and a certificate. You use a key management utility to generate a self-signed certificate and private key, and then you extract the self-signed certificate and add it to a *truststore* of the communicating application to serve as the CA certificate. Note that you do not include the private key in the file when you extract a self-signed certificate.

The Tivoli Identity Manager Server takes advantage of the WebSphere Application Server SSL keystore and SSL truststore for the SSL communication with adapters:

- ▶ The *SSL keystore* is a key database file designated as a keystore that contains the SSL certificate.
- ▶ The *SSL truststore* is a key database file designated as a truststore that contains the list of signer certificates (CA certificates) that define which certificates the SSL protocol trusts.

In the other words, only a certificate issued by one of these listed trusted signers is accepted.

Note that the keystore and truststore can be the same physical file but this practice usually is not the best security practice.

If enabling SSL communication between the Tivoli Identity Manager Server and an adapter, the configuration of Tivoli Identity Manager Server does not require any specific parameters. The only setup that is required on the Tivoli Identity Manager Server side is to configure the keystore and truststore properly in WebSphere Application Server.

If the Tivoli Identity Manager Server has to communicate to the directory server using LDAP over SSL, in addition to creating appropriate keystore and truststore in WebSphere Application Server, you need to update the Tivoli Identity Manager properties file (called `enRoleLDAPConnection.properties`) with the following parameters:

```
java.naming.provider.url=<ldap host>:<ldap port>
java.naming.security.protocol=ssl
```

For instructions about how to enable SSL on the directory server, use the applicable directory server documentation or the documentation about security that is available in the IBM Tivoli Identity Manager Information Center:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

We discuss how to enable SSL for the adapter in 3.4, “Adapter installation and configuration” on page 93.

3.2.1 Certificate and key formats

Certificates and keys are stored in files with the following formats:

► .pem format

A *privacy-enhanced mail* (.pem) format file begins and ends with the following lines:

```
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
```

A .pem file format supports multiple digital certificates, including a certificate chain. If your organization uses certificate chaining, you can use this format to create CA certificates.

▶ .arm format

An .arm file contains a base-64 encoded ASCII representation of a certificate, including its public key, but not its private key. An .arm file format is generated and used by the IBM Key Management utility. You can specify this format to extract a self-signed certificate from the machine on which the self-signed certificate was generated to the machine that will use the self-signed certificate as the CA certificate.

▶ .der format

A .der file contains binary data. A .der file can be used only for a single certificate, unlike a .pem file, which can contain multiple certificates. You can specify this format to extract a self-signed certificate from the machine on which the self-signed certificate was generated to the machine that will use the self-signed certificate as the CA certificate.

▶ .pfx format (PKCS12)

A PKCS12 file is a portable file that contains a certificate (CA-issued certificate or self-signed certificate) and a corresponding private key. This format enables you to transfer the contents of a keystore to a separate machine. For example, you can create and install a certificate and private key using the IBM Key Management utility, export the certificate and key to a PKCS12 file, and then import the file into another keystore. This format is also useful for converting from one type of SSL implementation to a different one. However, as this format contains a private key it is considered a non-secure way for managing certificates and it is not a recommended approach for the certificate management.

3.2.2 SSL handshake

The two nodes that participate in SSL communication exchange signed digital certificates in the authentication phase also known as an *SSL handshake*. The following steps describe typical SSL handshake communication:

1. The client sends a client “hello” message that lists the cryptographic capabilities of the client (sorted in client preference order), such as the version of SSL, the cipher suites supported by the client, and the data compression methods supported by the client. The message also contains a 28-byte random number.
2. The server responds with a server “hello” message that contains the cryptographic method (cipher suite) and the data compression method selected by the server, the session ID, and another random number.⁷

⁷ Note that the client and the server must support at least one common cipher suite, or the handshake fails. The server generally chooses the strongest common cipher suite.

3. The server sends its digital certificate (for example, the server uses X.509 V3 digital certificates with SSL). If the server uses SSL V3 and if the server application (for example, the Web server) requires a digital certificate for client authentication, the server sends a “digital certificate request” message. In the “digital certificate request” message, the server sends a list of the types of digital certificates supported and the distinguished names of acceptable certificate authorities.
4. The server sends a server “hello done” message and waits for a client response.
5. Upon receipt of the server “hello done” message, the client verifies the validity of the server’s digital certificate and checks that the server’s “hello” parameters are acceptable. If the server requested a client digital certificate, the client sends a digital certificate, or if no suitable digital certificate is available, the client sends a “no digital certificate” alert. This alert is only a warning, but the server application can fail the session if client authentication is mandatory. The negotiation that requires the client to provide a certificate to the server is called *two-way SSL*, or *mutual SSL*.
6. The client sends a “client key exchange” message. This message contains the pre-master secret, a 46-byte random number used in the generation of the symmetric encryption keys and the message authentication code (MAC) keys, encrypted with the public key of the server. If the client sent a digital certificate to the server, the client sends a “digital certificate verify” message signed with the client’s private key. By verifying the signature of this message, the server can explicitly verify the ownership of the client digital certificate.

Note: An additional process to verify the server digital certificate is not necessary. If the server does not have the private key that belongs to the digital certificate, it cannot decrypt the pre-master secret and create the correct keys for the symmetric encryption algorithm, and the handshake fails.

7. The client uses a series of cryptographic operations to convert the pre-master secret into a master secret, from which all key material required for encryption and message authentication is derived. Then, the client sends a “change cipher spec” message to make the server switch to the newly negotiated cipher suite. The next message sent by the client (the “finished” message) is the first message encrypted with this cipher method and keys.
8. The server responds with a “change cipher spec” and a “finished” message of its own.
9. The SSL handshake ends, and encrypted application data can be sent.

3.3 Installation process

Because the Tivoli Identity Manager installation includes several components, you need to plan the installation thoroughly, especially in larger environments.

Note: For a detailed description of the installation process, refer to *IBM Tivoli Identity Manager Server Version 5.0 Installation and Configuration Guide*, SC32-1562. For the latest prerequisites in versions and fix packs, consult the *IBM Tivoli Identity Manager Version 5.0: Product overview*, Release Notes section.

We discuss upgrade options in Chapter 8, “Maintenance” on page 205.

For helpful worksheets that contain all the questions that need to be answered during the configuration of middleware components and Tivoli Identity Manager Server, refer to “Appendix E Worksheets” in the *IBM Tivoli Identity Manager Server Version 5.0 Installation and Configuration Guide*, SC32-1562. We recommend that you review and use these sheets during your installation process. Those worksheets can be very useful for documentation purposes, too.

The following steps provide a high-level overview of how to install, configure, and test Tivoli Identity Manager:

1. Determine the Tivoli Identity Manager Server topology.

You can install Tivoli Identity Manager components as shown in Figure 3-1 on page 80 on a single server or distributed across several servers. Customer requirements and architectural decisions dictate the target environment.

2. Check for operating system prerequisites.

As part of this step, you need to ensure that the person who performs the installation has the necessary administrative rights on the system. A UNIX-based installation needs to be performed using the root account, and a Windows installation needs to be performed using the Administrator account.

3. Install and preconfigure the supported database server.

The database needs to be installed and configured before the Tivoli Identity Manager Server installation starts, because the new database is used by the Tivoli Identity Manager Server installation program, which populates the database with data objects.

4. Install and preconfigure the supported directory server.

The directory server needs to be installed and a company suffix configured before the Tivoli Identity Manager Server installation starts because the directory server is used by the Tivoli Identity Manager Server installation

program, which populates the Tivoli Identity Manager configuration under the created company suffix.

The Tivoli Identity Manager installation product includes a middleware configuration utility for DB2 and IBM Tivoli Directory Server. Other supported databases and SunOne Directory Server are not supported by this tool and need to be configured manually. Default values are supplied for many of the typical parameters and all of the advanced parameters. If an entered parameter such as the DB2 instance ID already exists, the middleware configuration utility will skip the task of creation. You can choose to keep those values, or you can provide values of your own. You can run the middleware configuration utility manually or silently by providing a response file as input.

The middleware configuration utility performs the following tasks for DB2:

- Creates user IDs if needed
- Creates DB2 instances if needed
- Creates databases if needed
- Tunes DB2 (bufferpool, log tuning)

The middleware configuration utility performs the following tasks for Tivoli Directory Server:

- Creates user IDs if needed
- Creates Tivoli Directory Server instances if needed
- Creates directory server databases if needed
- Tunes LDAP (bufferpool, log tuning)
- Adds the LDAP suffix
- Configures the non-SSL port
- Copies and configures the referential integrity plug-in

The middleware configuration utility has different names for the different platforms. For example, the name on System x® Linux is `cfg_itim_mw_xLinux`.

5. Optionally, install and preconfigure Tivoli Directory Integrator.

A Tivoli Identity Manager system can operate without Tivoli Directory Integrator in some occasions, but those occasions are very rare. Tivoli Directory Integrator is used to configure complex HR identity feeds into the Tivoli Identity Manager Server, as well as host agentless adapters based on RMI. Also, the majority of custom built adapters are based on the Tivoli Directory Integrator solution.

You can install Tivoli Directory Integrator on the same system as Tivoli Identity Manager or remotely. If you install Tivoli Identity Manager locally, the Tivoli Identity Manager installation program installs the agentless adapters automatically, and you can also choose to install automatically agentless adapter profiles that ship with the product. If you install Tivoli Identity Manager

remotely, you must manually install the agentless adapters on the computer that hosts Tivoli Directory Integrator and manually install the agentless adapter profiles on the computer that hosts the Tivoli Identity Manager Server by importing the profile .jar files or by using the `install` command.

For example, the following command installs an LDAP profile into the Tivoli Identity Manager Server:

```
ITIM_HOME\bin\win\config_remote_services.cmd -profile LdapProfile
-jar LdapProfile.jar
```

By default, the Tivoli Identity Manager installation program ships with the following agentless adapter service types (imported profiles):

- AIX profile (UNIX adapter)
- Solaris profile (UNIX adapter)
- HP-UX profile (UNIX adapter)
- Linux profile (Linux adapter)
- LDAP profiles (LDAP adapter)

6. Install the supported level of WebSphere Application Server. Depending on your choice of using the single server or cluster installation, you need a different type of WebSphere Application Server installation.

For the cluster installation, Tivoli Identity Manager assumes that the operating system is the same for each cluster member. Further, in a cluster, the name of the Tivoli Identity Manager installation directory must be the same for all cluster members.

7. Install the Tivoli Identity Manager Server application onto WebSphere Application Server.

The installation program is platform specific and has different names. For example for IBM System z Linux installation the name of installation wizard is `instzlinux.bin`.

Installation of Tivoli Identity Manager Server is more complex in the cluster environment than the stand-alone WebSphere Application Server. You need to perform the following steps in the cluster environment:

- a. You need to install Tivoli Identity Manager Server on the computer that hosts the WebSphere Application Server Deployment Manager before you install the Tivoli Identity Manager Server on cluster nodes. The deployment of the Tivoli Identity Manager application and the configuration of the database and the directory server for Tivoli Identity Manager occur during this installation. The Deployment Manager distributes and expands the Tivoli Identity Manager application to all cluster member computers.

- b. When the installation is complete on the Deployment Manager, you need to repeat the installation steps on the each cluster member. The installation program performs the following tasks:
 - i. Copies additional Tivoli Identity Manager files to the target computer.
 - ii. Configures the WebSphere Application Server that hosts the cluster member.

You must install the Tivoli Identity Manager Server on clusters sequentially, one cluster member at a time. Running the Tivoli Identity Manager installation program simultaneously on more than one computer at a time might result in synchronization problems with the WebSphere Application Server master configuration file.

Note: If the same computer hosts both the WebSphere Application Server Deployment Manager and a Tivoli Identity Manager cluster member, you must select both the Deployment Manager and the cluster member node types when you run the Tivoli Identity Manager installation program.

8. Configure the database, the directory server, and WebSphere Application Server for Tivoli Identity Manager Server.

The Tivoli Identity Manager installation program runs the *runConfig* system configuration tool automatically to edit commonly-used system properties for the Tivoli Identity Manager Server and also to configure WebSphere Application Server settings for the Tivoli Identity Manager application.

You can run this tool manually after the initial installation for updates of commonly-used Tivoli Identity Manager properties.

9. Test Tivoli Identity Manager Server and resolve any problems that occur.

After the installation and configuration, it is important that you verify the success of the installation and configuration with the following steps.

1. Verify that WebSphere Application Server is running by logging in to the WebSphere administrative console. You can do that by typing the following address in a Web browser:

```
http://hostname:port/ibm/console
```

For the value for *hostname*, use the server name where WebSphere Application Server is running. The value 9060 is the default port number for the WebSphere administrative HTTP transport. If you have multiple instances of WebSphere Application Server on the same computer, the port number might be different.

2. Check the Tivoli Identity Manager bus and messaging engine.

Before starting the Tivoli Identity Manager Server, use the WebSphere Application Server administrative console to check the status of the bus and messaging engine. To check the bus and messaging engine, complete the following steps.

- a. Start the WebSphere administrative console.

`http://hostname:port/ibm/console`

- b. Click **Service Integration** → **Buses**. If the bus is set, you see the `itim_bus`.

- c. Click **itim_bus**.

- d. In the Topology section, click **Messaging engines**.

For a single-server installation, you should see an engine named `nodename.servername-itim_bus`, and the status of the engine should be *started*.

For a cluster installation, you should see $n+1$ messaging engines, where n is the number of Tivoli Identity Manager cluster members, and an additional messaging engine is used for the Tivoli Identity Manager messaging cluster. All of these engines should be started.

3. On the WebSphere administrative console, click **Applications** → **Enterprise Application** and verify that the Tivoli Identity Manager Server *ITIM application* is running.

4. Log in to the Tivoli Identity Manager Server using the WebSphere embedded HTTP transport with the following command:

`http://hostname:9080/itim/console`

5. After successfully logging in to Tivoli Identity Manager Server using the WebSphere embedded HTTP transport, attempt to log in to the Tivoli Identity Manager Server using the IBM HTTP Server. Log with the following address:

`http://http_server_hostname/itim/console`

For the value of `http_server_hostname`, enter the host name of the IBM HTTP Server.

6. Verify that database and directory servers are running by using the appropriate administrative tools.

When this verification is complete, you can install appropriate adapters and begin the configuration of the product.

3.4 Adapter installation and configuration

After the successful installation of Tivoli Identity Manager Server, you continue with the following activities:

- ▶ Installing and configuring the adapters
- ▶ Optionally, configuring SSL communication between server and adapter
- ▶ Defining the corresponding service in Tivoli Identity Manager,
- ▶ Testing the communication between the adapter and Tivoli Identity Manager Server

Usually, the adapter performs tasks such as creating accounts, suspending accounts, and modifying account attributes. Therefore, an adapter can be considered a remote trusted virtual administrator on the target platform for account management, and we highly recommend that you configure the communication between the server and adapter to use SSL communication.

As already mentioned, Tivoli Identity Manager supports agentless and agent-based adapters. You install and configure these two types of adapters differently.

In securing communication with adapters, SSL is not the only option. Remote management of UNIX servers is performed using the SSH protocol based on Public Key Cryptography. The UNIX-like adapters use the same principle, and due to security reasons, SSH between the adapter and the managed resources cannot be disabled.

Each adapter provides its own installation and configuration guide that covers installation and configuration procedures for the adapter. The installation process consists of two general steps and applies to any type of adapter:

- ▶ Installing the adapter code on a remote system
For agent-based adapters, installing adapter code on a remote system actually implies the installation of adapter code on the managed resource. For agentless types of adapters, the adapter code is separate from the managed resource with which it is designed to communicate. It is installed on the system that hosts the Tivoli Directory Integrator system.
- ▶ Installing the service definition file (also known as *adapter profile*) on the Tivoli Identity Manager Server

Both types of adapters require that you install an adapter profile on the Tivoli Identity Manager Server using the **Import** → **Service Definition File** function in the Tivoli Identity Manager Web Administration Interface under **Configure System** → **Manage Service Types**.

The service definition file is a Java archive (.jar) that contains the following information:

- ▶ Service information, including definitions of the account provisioning operations that can be performed for the service, such as add, delete, suspend, or restore.
- ▶ Service provider information, which defines the underlying implementation of how the Tivoli Identity Manager Server communicates with the managed resource.
- ▶ Schema information, including the LDAP classes and attributes.
- ▶ Account forms and service forms, along with the label for the attributes, which are displayed in the user interface for creating services and requesting accounts on those services.

3.4.1 RMI-based adapters

RMI-based adapters require Tivoli Directory Integrator and the RMI Dispatcher component to run successfully. Figure 3-2 shows the RMI-based adapter architecture.

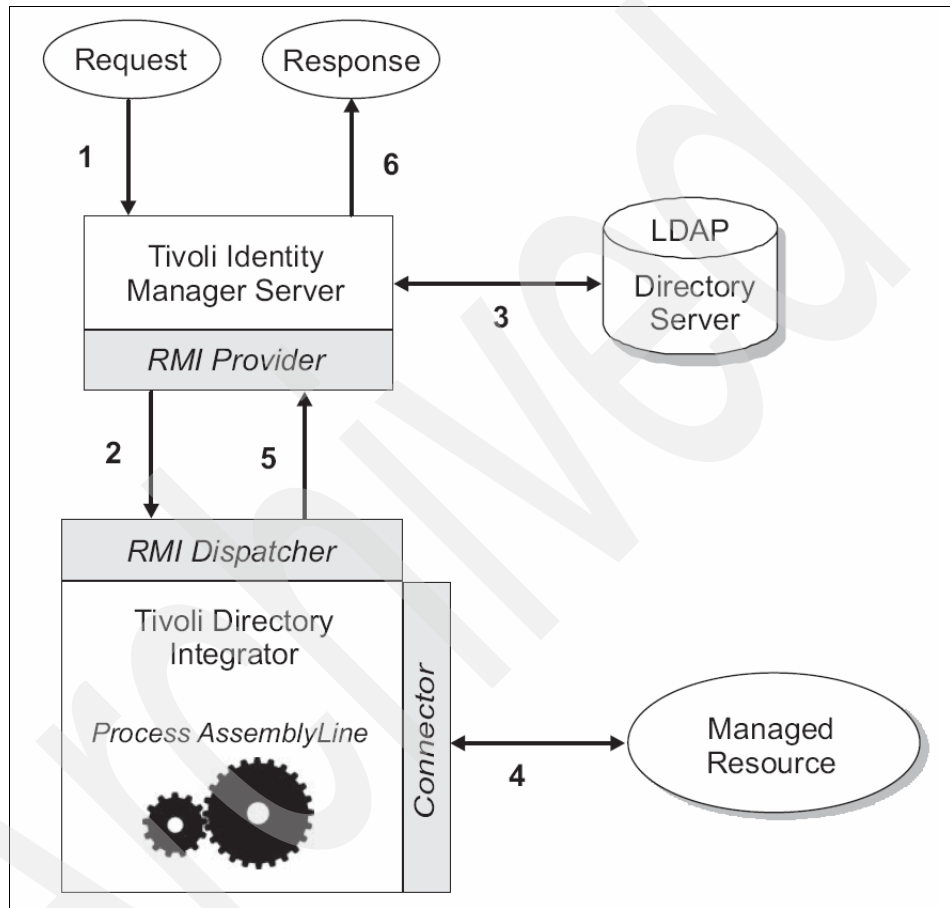


Figure 3-2 RMI adapter architecture

The following steps describe the information flow in this architecture:

1. A request for a task, or operation, to be performed for the managed resource is initiated with Tivoli Identity Manager.

Components of a customized adapter include an `AssemblyLine`, which contains the appropriate operations and connectors to communicate with the managed resource. The `AssemblyLine` normally supports the Add, Delete, Modify, Test, and Search operations.

2. The RMI Dispatcher on the Tivoli Directory Integrator receives the request from the RMI provider. Tivoli Directory Integrator expects to use an appropriate AssemblyLine to perform the requested operation. If the correct AssemblyLine is currently cached by Tivoli Directory Integrator, then it is used.⁸ If the correct AssemblyLine is not present, it is downloaded from the LDAP directory server (see step 3).

The RMI provider is included as part of the Tivoli Identity Manager Server. The RMI Dispatcher is a separate installation as an add-on to Tivoli Directory Integrator.

3. If required, Tivoli Identity Manager downloads the appropriate AssemblyLine for the requested operation and passes it using RMI to Tivoli Directory Integrator. The AssemblyLine is one of the components included in the custom profile for the adapter. Profiles are normally stored in the LDAP directory server that supports the Tivoli Identity Manager system.
4. Tivoli Directory Integrator invokes the AssemblyLine and uses one or more appropriate connectors (provided in Tivoli Directory Integrator) to communicate with the managed resource. The requested operation is performed for the managed resource and the result is returned to Tivoli Directory Integrator.
5. Tivoli Directory Integrator returns the result of the operation using RMI to the Tivoli Identity Manager Server.
6. If necessary, the Tivoli Identity Manager Server provides an appropriate response to the request.

For customization and configuration of RMI-based adapters, you must use the Tivoli Directory Integrator development interface.

You can also use the Adapter Development Toolkit (ADT), which is delivered separately from the Tivoli Identity Manager installation.⁹ The ADT provides the following functions:

- ▶ A graphical development environment that integrates Tivoli Directory Integrator functionality and Tivoli Identity Manager profile development.
- ▶ Reduced errors caused by manually editing files.
- ▶ Automated validation to identify common errors.
- ▶ Templates for adapter customization.
- ▶ Exporting and importing of adapters in either DSML or RMI formats.

⁸ Often during the development phase of a custom adapter, it is recommended to restart the RMI Dispatcher to clean the old AssemblyLine from the cache if a new version of profile is loaded into Tivoli Identity Manager.

⁹ You can download the ADT from the IBM OPAL Web site at:

<http://www.ibm.com/software/brandcatalog/portal/opal/details?catalog.label=1TW10IMOH>

Configuring SSL for RMI-based adapters

If SSL communication has to be enabled for the RMI-based adapter, the configuration requires that you create the appropriate SSL keystore and truststore and that you change the configuration of the `solution.properties` file in the RMI Dispatcher solution directory.

Edit the following two lines in the `solution.properties` file, depending on the type of secure communications that you want to use:

- ▶ For no SSL:

```
com.ibm.di.dispatcher.ssl=false  
com.ibm.di.dispatcher.ssl.clientAuth=false
```

- ▶ For one-way SSL:

```
com.ibm.di.dispatcher.ssl=true  
com.ibm.di.dispatcher.ssl.clientAuth=false
```

- ▶ For two-way SSL:

```
com.ibm.di.dispatcher.ssl=true  
com.ibm.di.dispatcher.ssl.clientAuth=true
```

3.4.2 ADK-based adapters

ADK-based adapters come with an installation wizard that is executed on the target managed resource. When installed, you configure the agent-based adapters on the remote system using the `agentCfg` command line tool as follows:

```
agentCfg -agent <adapter>Agent
```

Figure 3-3 shows the possible adapter configuration options when you issue the **agentCfg** command on an Active Directory Windows system.

```
Enter configuration key for Agent 'ADAgent': |  
  
      ADAgent 5.0.1000 Agent Main Configuration Menu  
      -----  
  
      A. Configuration Settings.  
      B. Protocol Configuration.  
      C. Event Notification.  
      D. Change Configuration Key.  
      E. Activity Logging.  
      F. Registry Settings.  
      G. Advanced Settings.  
      H. Statistics.  
      I. Codepage Support.  
  
      X. Done  
Select menu option:
```

Figure 3-3 agentCfg tool main menu for Windows adapter

Configuring SSL for ADK-based adapters

ADK-based adapters use DAML as the only protocol that supports SSL. To enable SSL communication (which is disabled by default) for the DAML protocol, you use the **agentCfg** tool. Figure 3-4 shows the DAML protocol properties.

```
DAML Protocol Properties
-----
A. USERNAME          ***** ;Authorized user name.
B. PASSWORD          ***** ;Authorized user password.
C. MAX_CONNECTIONS  100      ;Max Connections.
D. PORTNUMBER        45580    ;Protocol Server port number.
E. USE_SSL           FALSE    ;Use SSL secure connection.
F. SRV_NODENAME      _____ ;Event Notif. Server name.
G. SRV_PORTNUMBER    9443     ;Event Notif. Server port number.
H. HOSTADDR          ANY      ;Listen on address < or "ANY" >
I. VALIDATE_CLIENT_CE FALSE   ;Require client certificate.
J. REQUIRE_CERT_REG  FALSE    ;Require registered certificate.

X. Done

Select menu option:
```

Figure 3-4 Example of the DAML protocol properties

The parameter that enables SSL communication is under option E. If the setting `USE_SSL` is set to `TRUE`, you must use `certTool` to install the proper certificate.

In addition, if you want to use mutual-SSL, the I option needs to be set to `TRUE`. The property name actually is `VALIDATE_CLIENT_CERT`, and it is truncated by `agentCfg` to fit onto the screen. To use this option, you must use `CertTool` to install the appropriate CA certificates and optionally to register the Tivoli Identity Manager Server certificate.

As mentioned previously, on the adapter side, you use the same `certTool` command for managing certificates as follows:

```
certTool -agent <adapter>
```

Figure 3-5 shows the options that you need when you execute the **certTool** command.

```
IBM Tivoli Agent DAML Protocol Certificate Tool 4.63
-----
Main menu - Configuring agent: ADAgent
-----

A. Generate private key and certificate request
B. Install certificate from file
C. Install certificate and key from PKCS12 file
D. View current installed certificate

E. List CA certificates
F. Install a CA certificate
G. Delete a CA certificate

H. List registered certificates
I. Register certificate
J. Unregister a certificate

K. Export certificate and key to PKCS12 file

X. Quit

Choice:
```

Figure 3-5 certTool main menu for Windows adapter

This concludes the installation and configuration information for Tivoli Identity Manager adapters.

In the next chapter, we discuss the required configuration settings of the components that are necessary to implement Tivoli Identity Manager.



Implementation

In this chapter, we discuss the required configuration settings of the components necessary to implement IBM Tivoli Identity Manager. This chapter covers similar topics as in Chapter 2, “Planning” on page 39, but here, we provide more technical details of the Tivoli Identity Manager components.

4.1 IBM Tivoli Identity Manager components overview

After a successful installation of the Tivoli Identity Manager environment, you can continue with the configuration and implementation of the organization tree, services, policies, workflows, user groups and ACIs, reporting, and other system parameters. At this point, it is very important to understand the Tivoli Identity Manager components and their dependencies (for example the directory server tree structure, configuration files, and their organization on the disk).

Figure 4-1 shows all major Tivoli Identity Manager components and their relationships. As shown in the figure, the major relationship is between the identities and resources linked through authorization components.

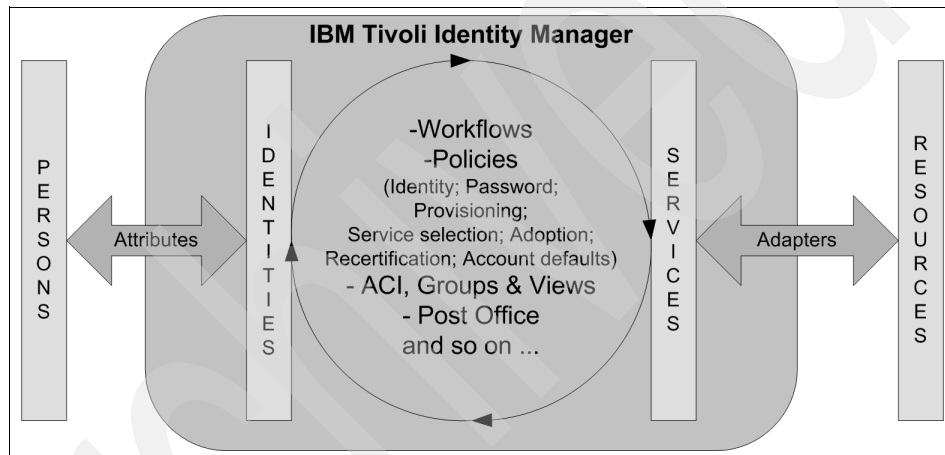


Figure 4-1 Dependencies between Tivoli Identity Manager components

We discuss the following main features (configuration elements) of Tivoli Identity Manager in this chapter:

- ▶ Organization tree
- ▶ Tivoli Identity Manager user types
- ▶ Services
- ▶ Policy
- ▶ Workflows
- ▶ Tivoli Identity Manager groups
- ▶ Access control item
- ▶ Views
- ▶ Auditing
- ▶ Reporting
- ▶ Post office
- ▶ Configuring commonly used system properties

- ▶ Modifying system properties manually
- ▶ Modifying system properties with the GUI
- ▶ User interface customization
- ▶ Directory server

4.2 Organization tree

The organization tree (org tree) is a useful functional representation of the organizational structure of the enterprise. The organization tree is created when the Tivoli Identity Manager system is first configured, and it is an important part of user and resource management because it enables you to define all the tiers of the organization.

Note: It is not necessarily the goal to find a most perfect real-life representation of an enterprise's organizational structure, but rather to determine the best way to manage person entities and resources throughout the entire organization. Some org trees might end up in a geographical representation, some might present the structure with only two separate branches for services and for users, and some might use a department structure regardless of where people are located. Consult *Identity Management Design Guide using IBM Tivoli Identity Manager*, SG24-6996, which discusses the organization chart design in more detail.

There is no one-way approach when designing and creating an organization tree structure. During the discovery and design phases of the implementation, a number of items can help determine the appropriate structure. The general approach is to keep the structure as simple as possible, because after most objects are created within an organizational container, you have to delete and re-create them to move them into a different part of the organization tree. However, you can move existing user objects to a different business unit by using a transfer.

If the organization tree structure is strongly tied to the company structure, any change in the company (such as a re-organization or a merger) can create a lot of administrative overhead for the identity management system.

When the organization tree is designed, you can attach different entities (people, policies, Tivoli Identity Manager groups, ACIs, roles, and so on) to different parts of the tree and to any node within the tree. However, the position in the tree (and the type of the entity) dictates the *scope* of the entity. For example, an *identity policy* can impact the services that are associated within the same business unit or the subtrees below the business unit. A different example is the *static role*,

which has an impact on every user in the entire organization to which the role belongs.

4.2.1 Organization tree elements

An *organization tree* consists of the following elements:

- ▶ Organization, at the top level
- ▶ Organizational units
- ▶ Business partner organizations
- ▶ Locations
- ▶ Admin domains

The organization tree always starts at the top level with an *organization* node. The next levels can be any combination of *organizational units*, *locations*, *business partner organizations*, and *admin domains*. The organization tree can have as few or as many levels of hierarchy that are required to suit the needs of the company. You can add, delete, or modify subsidiary items as the organization's structure changes in the future.

After the tree is built, you can populate different entities all over the tree. However, in this release, you cannot browse and create entities by navigating the organization tree. The association to a business unit within the organization tree is specified during the creation of the entity, or through the *placement rules*. Also, the users are located using the menu task *Manage users* to search for specific users based on different attributes and LDAP filters that can be specified in the Advanced Search panel.

When Tivoli Identity Manager users are added to the system, you have to make sure that they are placed in the correct branch. All people are attached to the organization tree at a single point.

4.2.2 Organizational roles

The *organizational roles* are used to model and represent job roles within an organization. They can be used to map users to a set of accounts that are granted through a *provisioning policy*. Depending on the nature of the organization and the complexity of the organization tree, several organizational roles can be created to suit the needs of the organization.

By placing a user in an organizational role, that user automatically can be granted access to managed resources in the organization. However, before the user can access the resource, that user must be provisioned with an account on the resource (service). In order for a person to be provisioned with an account, the organizational role must be a member of a provisioning policy.

You can use two organizational roles:

- ▶ Static organizational roles
- ▶ Dynamic organizational roles

Static organizational roles

Static organizational roles are available globally to any user of the organization. Assigning a user to a static organizational role can be done when:

- ▶ A user is added to Tivoli Identity Manager initially (either manually through the Web interface or automatically through a data feed).
- ▶ An existing user profile is modified.

Dynamic organizational roles

Dynamic organizational roles use an LDAP filter to automatically assign members based on any particular attribute found in a user's Tivoli Identity Manager profile. These attributes can be any type of profile information, such as title, address, employee number, or department name. Implementation of dynamic organizational roles need to be carefully analyzed because in some cases an LDAP filter can result in exhaustive LDAP searches and effect the performance of the overall Tivoli Identity Manager system.

Users obtain memberships to dynamic organizational roles when:

- ▶ The dynamic organizational role is created and the information for a Tivoli Identity Manager user contains an attribute value targeted by the LDAP filter.
- ▶ An existing user's profile information is updated and the updated attributes contained in the user's profile match the information in the definition (rule) of the role. You use LDAP filters to define these rules. These rules can either prequalify or automatically provision the user with any resources that are associated with the entitlement of a provisioning policy.

4.2.3 Placement rules

Most Tivoli Identity Manager installations require a bulk load process in order to load person data into the system, mostly from human resources systems. The Tivoli Identity Manager data load mechanism can include JavaScript to define where each new user is placed in the tree. If you do not include a *placement rule*, all users are placed into a default organization, and you must later move each user into the correct organization container.

Placement rules return the organization path in a distinguished name (DN) format. This information is used to search for an organizational unit in which to

place a person. This DN indicates the required organization path relative to the organization base.

The syntax of this path can be represented with the following pseudo Backus-Naur Form notation (BNF syntax):

```
orgDn ::= orgRdn | orgRdn "," orgDn
orgRdn ::= prefix '=' name
prefix ::= 'l' | 'o' | 'ou'
name ::= string
```

In this syntax:

- ▶ *string* is the textual value
- ▶ *l* is location
- ▶ *o* is organization
- ▶ *ou* is the organizational unit, business partner organization, or Admin Domain

The Tivoli Identity Manager Server evaluates this script when adding a new person to determine where to place that person in the organization. During a modify request, this script is evaluated and, if the value is different than the current placement of the person, the person is moved to the new location based on the returned path.

4.3 Tivoli Identity Manager user types

Tivoli Identity Manager operates with three different types of users:

- ▶ Person
- ▶ Business partner person
- ▶ Custom person

Depending on how Tivoli Identity Manager is configured, the system can add users automatically using information from a human resources database or other sources that it receives during a data feed operation. However, there might be instances where you need to add these users manually into Tivoli Identity Manager. It is also possible for users to add themselves through a predefined self-registration process.

A *person* represents an employee of a managed organization or company.

A *business partner person* (BPPerson) is an individual who is not an employee of the organization but who needs access to resources that are managed by Tivoli Identity Manager. Business partner persons usually represent contractors and consultants who work for outside organizations.

Person and BPPerson are categories that are built into Tivoli Identity Manager. Both classes are managed the same way, but when adding a BPPerson, less information is required.

Users can be located anywhere in the organization tree, but when adding a new user, you must select the appropriate branch of the organization where to add the new user. The personal information is stored as attributes to the person objects, which can include first, last, and full names, phone numbers, employee number, supervisor, e-mail address, and so on. The person with its attributes corresponds to the standard LDAP object class `iNetOrgPerson`.

If you need to store additional attributes for your person that are not defined in `iNetOrgPerson`, you can create new LDAP object classes with custom attributes by using appropriate LDAP tools or commands. After a new object class is created, you can use the Tivoli Identity Manager administration interface to define a new *custom person* under **Configure System** → **Manage Entities**. A part of this customization is to define a default search attributes list. If attributes are selected to be used for the extensive and frequent search operation, recommendation is to create indexes on those attributes in LDAP to optimize performance.

For every new custom person, you need to define separate LDAP object class.

When a BPPerson is registered in Tivoli Identity Manager, a *sponsor* can be selected. A sponsor is usually the manager of the business partner person. You choose a BPPerson's sponsor just as you choose a manager (supervisor) for a person (employee).

4.4 Services

A *service* represents an instance of a managed resource that a user can subscribe or be provisioned to in order to be granted access. For a user to gain access to a service, you need to define a *provisioning policy* in order to create and maintain an account on that resource. We discuss the provisioning policy in 4.5.3, “Provisioning policy” on page 119. An adapter needs to be installed and a service configured to communicate with the adapter in order to manage users on the resource.

A service is created from a *service type* (also known as an *adapter profile*) that can be considered a template with a common set of attributes for each type of managed resource supported by Tivoli Identity Manager. Many service types are shipped with Tivoli Identity Manager or they can be imported from a .jar file as discussed in 3.4, “Adapter installation and configuration” on page 93.

Tivoli Identity Manager comes with following default service types:

► Identity feed service types

Those service types do not create accounts. They are used to import user data from an authoritative data source of identities into the Tivoli Identity Manager directory as Person information. The following identity feed service comes with Tivoli Identity Manager:

- Directory Service Markup Language (DSML) identity feed
- Active Directory
- iNetOrgPerson
- Comma-separated value (CSV) file
- IDI data feed¹

► Account service types:

- IBM Tivoli Directory Integrator based adapters (LDAP, UNIX, and Linux)
- IBM Tivoli Identity Manager Service
- Hosted service
- Custom Java class
- Manual service

4.4.1 Identity feed service types

A *DSML Identity Feed* is the service type that you use to import user data from a DSML file. A DSML file represents directory structure information in an XML file format. The DSML file must contain only valid attributes of the Tivoli Identity Manager profile. The identity feed process uses all objects in the file. When a DSML feed based service is created, it does not create persons automatically into Tivoli Identity Manager. The service can receive the information using a reconciliation process or an unsolicited event notification through an event notification program.

An *AD Identity Feed Service* imports user data from Windows Active Directory. The Active Directory presentation of the user is based on the `organizationalPerson` objects class. Thus, the user profiles that are selected from this service must have an objectclass that is derived from the `organizationalPerson` class, or the attribute mapping file is created that maps to attributes that come with the standard `inetOrgPerson` object class. The attribute mapping file completely overrides the default mappings. All attributes that are needed from the feed source must be included in the mapping file. If an attribute from the feed source is not included in the attribute mapping file, the value is not set on the Tivoli Identity Manager person attribute.

¹ Although the respective product behind the IDI data feed is renamed from IBM Directory Integrator to IBM Tivoli Directory Integrator. The service type name in Tivoli Identity Manager is not changed.

The *iNetOrgPerson Identity Feed* imports user data based on the standard user object class in LDAP directory - *iNetOrgPerson*. It works similar to the AD Identity Feed, except that the base is different object class. The *inetOrgPerson* objects are loaded and users are added or updated in Tivoli Identity Manager.

A *CSV Identity Feed Service* imports user data from a comma-separated value (CSV) file and adds or updates persons into Tivoli Identity Manager. CSV file contains a set of records separated by a carriage return/line feed (CR/LF) pair (`\r\n`). Each record contains a set of fields separated by a comma. If the field contains either a comma or a CR/LF, the comma must be escaped with double quotation marks as the delimiter. The first record in the CSV source file defines the attributes provided in each of the following records. Attributes must be valid based on the class schema for the selected person profile for this service.

A *IDI data feed* service type uses the Tivoli Directory Integrator to import identity (not account) information into Tivoli Identity Manager. IDI Data feed is provided for instances where the other identity feeds are not sufficient and provide greater flexibility over the standard, above mentioned, data feeds.

4.4.2 Account service types

Besides being heavily involved in the identity feed discipline, we find Tivoli Directory Integrator-based agentless adapters. For Tivoli Identity Manager, Tivoli Directory Integrator is just another type of service provider. This service type can be installed optionally during the installation of Tivoli Identity Manager. The LDAP and UNIX/Linux service types come with the Tivoli Identity Manager installation.

An *IBM Tivoli Identity Manager service* is a service that is provided with IBM Tivoli Identity Manager for creating Tivoli Identity Manager accounts. All users that need access to the Tivoli Identity Manager system must be provisioned with a Tivoli Identity Manager account. This is a standard service with no configuration parameters.

A *Hosted Service* is used to connect to services residing in other organizations or organizational units. The Hosted Service connects to the managed resource target through the hosting service indirectly. The configuration details of the hosting service are invisible and protected from administrators in the secondary organization where the Hosted Service is defined. Administrators can define policies for the Hosted Service, specifically, without affecting the hosting service. The primary usage of a Hosted Service allows users in business partner organizations to have accounts and access to internal IT resources of an organization and to allow administrators in the secondary organization to define specific service policies for the user accounts.

A *Custom Java class service type* allow you to define your own profile by defining and implementing a Java class.

A *Manual service* is a type of service that requires manual intervention to complete the request. For example, a manual service can be defined for setting up voice mail for a user. Manual services generate a workflow activity that defines the manual intervention that is required. You can create a manual service when Tivoli Identity Manager does not provide an adapter for a managed resource for which you want to provision accounts, or there is no business justification to invest into custom adapter development (for example, if the number of managed accounts is too low). When you create a manual service, you add new schema classes and attributes for the manual service to the Tivoli Identity Manager LDAP directory.

Other service types are added to the list and made available to the system when Tivoli Identity Manager adapters are installed. Remember, to create a new service, a service profile needs to be imported into Tivoli Identity Manager Server, as described in 3.4, “Adapter installation and configuration” on page 93.

4.4.3 Reconciliation

After the initial configuration of a service, we can invoke a *reconciliation* for that service. Reconciliation is the process of synchronizing user account information, along with its attributes, stored in Tivoli Identity Manager Server and account information stored on the remote system (resource). Reconciliation is required when accounts and supporting data can be changed on the managed resource so that Tivoli Identity Manager Server data is consistent and up-to-date with the remote resource. During the reconciliation process, Tivoli Identity Manager communicates with the adapter on the remote system, compares data, and takes appropriate action.

If there is a match between a user login ID and an account, Tivoli Identity Manager creates the ownership relationship between the account and the person. If there is no match, Tivoli Identity Manager marks the unmatched account as an *orphan account*.

There are two types of reconciliation processes:

- ▶ Load access information into the Tivoli Identity Manager

When a service is first brought into Tivoli Identity Manager for management of the managed resource’s accounts, there needs to be an initial load of the accounts and accompanying data associated with the service. This is performed by an initial reconciliation.

- ▶ Monitor accesses granted outside of Tivoli Identity Manager administration
Periodically, reconciliations need to be run to monitor the state of accounts and note if they have changed and no longer meet the policies defined within Tivoli Identity Manager. Accounts that are not owned by people (orphan accounts) are also monitored through these means.

Let us now take a closer look at the reconciliation process. If an entry for a managed account already exists within Tivoli Identity Manager, reconciliation simply updates the account's details within Tivoli Identity Manager, if necessary, or performs the corrections in the actual account on the managed endpoint. The decision about what needs to be done is made based on the Tivoli Identity Manager provisioning policy and the service definition.

For example, if a person's Microsoft Windows account has been modified since the last reconciliation by a source external to Tivoli Identity Manager and if that account has been made a member of a group called *HR*, the account record within the Tivoli Identity Manager directory can then be updated in the account record within the Tivoli Identity Manager directory accordingly. If the relevant policy states, however, that the person should not be a member of any groups within Windows, Tivoli Identity Manager will either mark the account as non-compliant, suspend the person's Windows account, alert the relevant person that the account is non-compliant and route an activity to their to-do list, or correct the Windows account to no longer be a member of the HR group within Windows.

An account created outside of Tivoli Identity Manager, for example, a local UNIX root administrator creates a new ID with root access rights, might not be acceptable according to defined provisioning policies. Enabling *policy checking* during reconciliation enables you to identify areas in your organization that are not compliant with security policies and to take appropriate actions defined through policy enforcement.

Reconciliation is defined per service and can be scheduled, or initiated a immediately. Reconciliations are resource-intensive operations that require a lot of server memory (Java Virtual Machine, JVM™ memory) and can take a while for services with a large account population. Because any change to the account will trigger the policy evaluation for that account regardless if the change would invalidate the policy, you should consider limiting the number of attributes returned by the adapter and processed by Tivoli Identity Manager for performance reason. To help with that, LDAP filters can be setup on some adapters when configuring the reconciliation.

To optimize reconciliation, you can also separate reconciliation of accounts from the supporting data reconciliation. Supporting data includes group configuration information, which contains key information about access privileges on the

resource. Bringing back the group data ahead of time allows policies to be configured promptly before accounts are reconciled, so that the policies can be enforced.

Large reconciliations can also exceed the default Max Duration and if so, the value can be increased.

Adoption policy

The reconciliation process can generate what is known as orphan accounts. These are accounts to which Tivoli Identity Manager cannot assign owners.

An *adoption policy* is used during reconciliation to determine the owner of an account by using *adoption rules*. An adoption policy can be created either globally, per service type, or per service to specify how to adopt orphan accounts during a reconciliation. Service instances of different types cannot be defined in the same adoption policy. An adoption policy does not alter the ownership for accounts that already have an owner assigned within Tivoli Identity Manager.

Adoption policies are defined through the use of JavaScript. By default, a global script is supplied that adopts accounts by preferred user ID attribute of the person (*uid* LDAP attribute). This approach differs from the default policy in Tivoli Identity Manager 4.6. The comparison there was against aliases (*eraliases* LDAP attribute).

You need to reconcile your data on a scheduled basis for your organization's ongoing security audits.

The action to be triggered during *policy enforcement* is configured on the Service Policy Enforcement page. This page is accessible through the Policy Enforcement link in the Service menu.

Note: All services except the DSML identity feed services have policy enforcement available.

If the **Check Policy during Reconciliation** check box is selected for your reconciliation process, the system takes different actions on accounts depending on what provisioning policy *actions* are specified in the Service configuration:

- ▶ **Correct non-compliance:** Correct the non-compliances for the account. Looking back at our previous example, this would result in the account being removed from the "HR" group within Windows.
- ▶ **Suspend non-compliance:** Suspend the account.
- ▶ **Mark non-compliance:** Mark (flag) the account as non-compliant in Tivoli Identity Manager.

- ▶ Alert: Same as mark with the addition of sending a compliance alert to a desired participant (e-mail, to-do item, or both).
- ▶ Global Setting: Use the *global* Policy Enforcement Action setting. This enables you to specify the actual enforcement action (and associated customizing) in one place and have the services use whatever is specified there.

4.5 Policy

A policy represents a set of rules that influence the behavior of the resource targeted by the policy. Tivoli Identity Manager can help the organization to control the enforcement of different types of policies from a centralized location. Tivoli Identity Manager supports the following types of policies:

- ▶ Identity policy
- ▶ Password policy
- ▶ Provisioning policy
- ▶ Service selection policy
- ▶ Adoption policy
- ▶ Recertification policy
- ▶ Account defaults

4.5.1 Identity policy

Tivoli Identity Manager can manage the access rights for all computing platforms within an organization. With several thousands of users, it is possible that some users have the same name. Because most login IDs are generated using the first initial and last name of a user, there might be potential problems with duplicate login IDs. Therefore, you need to create a method to generate login IDs automatically, which take into consideration such situations. An identity policy defines how user IDs are created when requesting a new account on the resource managed by Tivoli Identity Manager. Based on the policy, login IDs can be created for every platform managed by Tivoli Identity Manager. Because each user must have a unique login ID for each resource they access, Tivoli Identity Manager can use the identity policy to take the guesswork out of the login ID creation process.

When you provision a user with a service that has an identity policy assigned, you will notice that the login ID are populated automatically into the login ID field. This automatic generation of login IDs indicates that the identity policy is working correctly.

Based on the needs of the organization, identity policies can be created globally for all services, or for specific services (a single service or an instance of a specific service). However, identity policies should be placed at the same level or higher in the org tree than the services to which they are applied, and the scope can be set to single level or subtree. As we all know, some platforms can have strict login ID creation requirements, such as having an ID that is not longer than a certain number of characters. Identity policies must all conform to the policies of each platform to avoid any conflicts in login ID generation.

Tivoli Identity Manager manages access for both persons and business partner persons. Therefore, identity policies must be created for each type of Tivoli Identity Manager user, which can be extremely useful if you use a different format for contractors versus employees.

When creating an identity policy, the information contained in the user registration is available to the policy for use. For example, you can use the simple first initial and last name format. Your script can then add a number to the end of this concatenated value if it finds a conflict. You can also use other information from the user record. If your registration feed includes an employee number, aid, or other value that you want to use as a login ID, you can use that as well. When writing your identity policy, be sure to include a call to verify that the login ID has not already been used to prevent the account creation request from failing.

In the process of creating an identity policy you can define a rule that specifies which attributes to use, how many characters to use from each attribute, and whether to modify the letter case (existing, all upper, or all lower case, which is default) when creating a user ID. The rule can be defined using a basic or an advanced approach. The basic approach requires no scripting, yet you can use it to define basic rules for each type of policy. The advanced approach involves scripting, and you can use it to define more complex and customized rules. Tivoli Identity Manager provides a default script you can modify.

Example 4-1 illustrates the advanced mode of creating user IDs using the uid attribute (or given name, if the uid attribute is empty) for an individual. The script also checks if the user ID is already used. If the user ID is already in use, the script adds a number to the end of the user ID, making it unique.

Example 4-1 Identity policy advanced mode JavaScript

```
function createIdentity() {
    var EXISTING_CASE = 0;
    var UPPER_CASE = 1;
    var LOWER_CASE = 2;
    var tf = false;
    var identity = "";
    var baseidentity = "";
```



```

var counter = 0;
var locale = subject.getProperty("erlocale");
var fAttrKey = "uid";
var sAttrKey = "";
var idx1 = 0;
var idx2 = 0;
var fCase = 2;
var sCase = 2;
if ((locale != null) && (locale.length > 0)) {
    locale = locale[0];
}
if (locale == null || locale.length == 0)
    locale = "";
var firstAttribute = "";
var secondAttribute = "";
if (((fAttrKey != null) && (fAttrKey.length > 0)) || ((sAttrKey !=
null)
&& (sAttrKey.length > 0))) {
    if ((fAttrKey != null) && (fAttrKey.length > 0)) {
        firstAttribute = subject.getProperty(fAttrKey);
        if (((firstAttribute != null) && (firstAttribute.length > 0)))
            firstAttribute = firstAttribute[0];
        if (firstAttribute == null || firstAttribute.length == 0)
            firstAttribute = "";
        else {
            firstAttribute = IdentityPolicy.resolveAttribute(fAttrKey,
            firstAttribute);
            if ((idx1 > firstAttribute.length) || (idx1 == 0))
                idx1 = firstAttribute.length;
            firstAttribute = firstAttribute.substring(0, idx1);
        }
    }
    if (fCase == UPPER_CASE)
        firstAttribute = firstAttribute.toUpperCase(locale);
    else if (fCase == LOWER_CASE)
        firstAttribute = firstAttribute.toLowerCase(locale);
}
if ((sAttrKey != null) && (sAttrKey.length > 0)) {
    secondAttribute = subject.getProperty(sAttrKey);
    if (((secondAttribute != null) && (secondAttribute.length > 0)))
        secondAttribute = secondAttribute[0];
    if (secondAttribute == null || secondAttribute.length == 0)
        secondAttribute = "";
    else {
        secondAttribute = IdentityPolicy.resolveAttribute(sAttrKey,
        secondAttribute);
        if ((idx2 > secondAttribute.length) || (idx2 == 0))
            idx2 = secondAttribute.length;
    }
}

```

```

        secondAttribute = secondAttribute.substring(0, idx2);
    }
    if (sCase == UPPER_CASE)
        secondAttribute = secondAttribute.toUpperCase(locale);
    else if (sCase == LOWER_CASE)
        secondAttribute = secondAttribute.toLowerCase(locale);
    }
    baseidentity = firstAttribute + secondAttribute;
}
if ((baseidentity == null) || (baseidentity.length == 0)) {
    var givenname = subject.getProperty("givenname");
    if ((givenname != null) && (givenname.length > 0))
        givenname = givenname[0];
    if (givenname == null || givenname.length == 0)
        givenname = "";
    else
        givenname = givenname.substring(0, 1);
    baseidentity = givenname + subject.getProperty("sn")[0];
}
tf = IdentityPolicy.userIDExists(baseidentity, false, false);
if (!tf) {
    return baseidentity;
}
while (tf) {
    counter+=1;
    identity = baseidentity + counter;
    tf = IdentityPolicy.userIDExists(identity, false, false);
}
return identity;
}
return createIdentity();

```

If the identity policy generates a user ID with a null value, Tivoli Identity Manager attempts to form a user ID by using the first letter of the user's given name, concatenated with the value of the user's family name, retaining the existing case.

4.5.2 Password policy

All accounts have passwords. Account passwords can be centrally managed by their owners or administrators using the Tivoli Identity Manager Web interface. Also, passwords can be synchronized. The synchronization can be applied to all accounts associated with a user or selected accounts. For most passwords, this is a one-way synchronization. Tivoli Identity Manager sets the password and pushes it to the managed targets. Tivoli Identity Manager cannot accept a

password change request from a target and push this to all associated accounts. The exception to this is the Microsoft Windows NT and Active Directory password synchronization function, the Reverse Password Synchronization for Tivoli Access Manager WebSEAL agent, which intercepts a password change and passes it through Tivoli Identity Manager and any LDAP with its changelog enabled.

A password policy sets parameters (password rules) that all passwords must meet, such as length and type of characters allowed and disallowed. Just as certain platforms have strict login ID creation requirements, these platforms can also have strict password requirements. Your organization might also have a strict password policy. Tivoli Identity Manager provides several built-in password rules that can be enabled or disabled when creating a password policy for a particular platform. In addition to the built-in password rules, a Tivoli Identity Manager administrator can create a new customized rule, a customized generator, or a combination of both.

A *customized rule* for generating passwords using the Tivoli Identity Manager Server can be accomplished by creating a Java class by implementing the `com.ibm.passwordrules.Rule` interface. The class must be registered in `passwordrules.properties`. Optionally, a label for the customized rule name can be added in `CustomLabels.properties`. If the customized label is not defined in `CustomLabels.properties`, the fully qualified name of the customized Java class is shown on the interface forms.

A *customized password generator* for creating passwords using the Tivoli Identity Manager Server can be accomplished by creating a Java class by implementing `com.ibm.passwordrules.PasswordGenerator` interface. Again, the class must be registered in `passwordrules.properties`.

After users are granted access, they can change their passwords in Tivoli Identity Manager. The new password must conform to the rules of the password policy. Users can view the rules of the password policy for each system by clicking the View Password Rules button.

Based on the needs of the organization, password policies can be applied for any of the following items:

- ▶ Only one service instance or more than one service instance
- ▶ All service instances of only one service type or multiple service types
- ▶ All services, regardless of service type

Similar to identity policies, a password policy should be placed at the same level or higher in the org tree than the services to which they will be applied, and the scope can be set as single level or subtree.

If a password policy exists for all services, other policies can still be added. However, only one password policy can be specified for each service type or each instance of a service type. If a password policy exists for a service type, as well as password policies for different instances of that service type, the more specific password policy overrides all others (for example, a password policy for a Windows AD service instance overrides a password policy for the Windows AD service).

It is possible to have two or more services that have very different password requirements. You might be able to come up with one password policy that will meet all of the requirements, but it might be too weak for your corporate security guidelines. In this case, you need to create a password policy that applies only to this “weak” service, and users will have to pick a different password for this specific service.

If users have accounts on multiple systems that have differing password policies, they can select all of their accounts and view the combined password rules. This will show the user what components will make a password that is acceptable across all systems. If the policies are very different, it is possible that there will be no common password rule that can work for the selected accounts. In that case, the user has to select them individually to change the passwords.

In addition, when password synchronization is enabled, Tivoli Identity Manager combines policies for all accounts that are owned by the user to determine the password to be used. If conflicts between password policies occur, the password might not be set.

Password dictionary

An *password dictionary* contains a list of words, stored in the directory server, that cannot be used as passwords.

This dictionary can be modified through an LDAP browser by creating `erDictionaryItem` entries under the `erDictionaryName=<password>` entry or by importing an LDIF file with the entries listed into the directory server. For more details, refer to the IBM Tivoli Identity Manager Information Center:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

After the password dictionary is populated with the desired words, the password policies must be modified to use the dictionary. Upon importing the LDIF file, select the “Do not allow in dictionary” option on the Rules page of password policies.

Forgotten password settings

Tivoli Identity Manager uses a Forgotten Password Settings challenge or response function to verify users' identities if they have forgotten their Tivoli Identity Manager passwords. The challenge questions can be:

- ▶ **User-defined:** The user can create a unique Password Challenge Response questions and answers.
- ▶ **Admin-defined:** The user can create Password Challenge Response answers to predefined questions that are determined by the Tivoli Identity Manager administrator.

When a user logs in to Tivoli Identity Manager for the first time, the user enters the challenge questions (if configured) and responses. On subsequent logins to Tivoli Identity Manager, the user can select a "Forgot password" option and a subset of the challenge responses are used to verify the user.

Also, Tivoli Identity Manager can be configured to respond to lost password behavior in two ways:

- ▶ **Reset and e-mail password**
Select this option to configure the system to send a new password to the e-mail address associated with the account.
- ▶ **Log in to system**
Select this option to configure the system to log the user in to the system.

Responses (answers) are saved in the directory server as non case-sensitive by default, if you want answers to be case-sensitive, change the value for `enrole.challengeresponse.responseConvertCase` from lower to upper. Also the answer can be preserved in original writing if the value is set to none.

4.5.3 Provisioning policy

Provisioning policies define the level of access a user is granted to a resource. In order to create a *provisioning policy*, you must define:

- ▶ **Membership:** Who is allowed access.
- ▶ **Entitlements:** What resources are provided to users.

When creating a provisioning policy, keep in mind that complicated provisioning policies can result in complicated directory and database queries with poor performance. Policies with a small numbers of roles and services will perform best.

Also, the scope of the provisioning policy depends on the position in the org tree and indicates whether to cover services in the same level of the business unit or the subtree of the business unit.

Membership

Provisioning policies are based on the following *membership* options:

- ▶ User membership to the organization (all people). In this case, any person registered in Tivoli Identity Manager will have access to the provisioned service.
- ▶ User membership to one or more organizational roles. In this case, a person must belong to a specific organizational role before they will be granted access to the provisioned service.
- ▶ User membership to no organizational role (others). This case enables you to provide services to people that do not belong to any organizational roles.

Certain services and organizational roles can have strict provisioning policies, such as those policies created for HR applications. However, policies created for services such as e-mail might not be as strict, especially because every user in the company typically requires access to e-mail.

Entitlements

An *entitlement* is part of a provisioning policy that determines which users are allowed to have access to certain resources. Entitlements are used to determine:

- ▶ What services are provided to the users defined in the policy membership.
- ▶ Whether the accounts are provisioned manually or automatically.
- ▶ How the account is configured (what attributes and values it has, such as default groups and home directory).

An entitlement in the provisioning policy supports different types of service targets, including all services, services of same type, services defined by service selection policy, or specific service instances.

Entitlements can be either automatic or manual. If a provisioning policy uses an automatic entitlement, a user will automatically be provisioned with services when the user meets the membership requirements. If a provisioning policy uses a manual entitlement, a Tivoli Identity Manager system administrator needs to take action to provision the user with the respective services.

Each service targeted by a provisioning policy can also have a *workflow* attached. A workflow is used to introduce additional processing before access is provisioned. A workflow can determine an approval process for a particular

service or set of services. For more information about workflows, see 4.6, “Workflows” on page 128.

Entitlement target type

Each provisioning policy is responsible for defining who gets accounts on one or more services. These services are called targets. The targets of a provisioning policy entitlement determine which accounts the policy members are entitled to receive.

There are four ways to configure the *target type* of a provisioning policy entitlement:

- ▶ All services, meaning that any service is available to a person that meets the membership requirements.
- ▶ By service profile, meaning that any services of a particular type are available to a person that meets the membership requirements.
- ▶ By a specific service instance name.
- ▶ By a service selection policy, which is a policy that determines a specific service based on the user information.

It is possible to test a provisioning policy by using the **Preview** button and check the results in order to avoid any undesirable changes to accounts and managed targets. The preview feature displays all account errors (non-compliances) and what actions would have taken place.

You can select one of two types of provisioning policy enforcements to check during the simulation:

- ▶ Changes only
Perform a provisioning policy simulation to preview the results for only the changes you are about to make.
- ▶ Entire policy
Perform a provisioning policy simulation to preview the results for all computed enforcement actions for the entire policy.

It is possible to save a draft of a provisioning policy. This way, you can avoid an immediate execution in a production environment in order to minimize the risks of a misconfigured provisioning policy. To determine the impact of a new or changed provisioning policy, a *provisioning policy simulation* can be run against a draft or a committed provisioning policy.

Let us take a look at a best practice approach:

1. Make changes to or create a new provisioning policy, and save it as a draft.

2. Run a provisioning policy simulation against the draft provisioning policy.
3. Commit the provisioning policy if everything works as expected. The action causes the draft to be deleted automatically.
4. Delete or reconfigure the draft if the simulation showed undesirable effects.

Sometimes, users have several provisioning policies that apply to them. When two or more provisioning policies are applied to the same user, a *join directive* defines how to handle attribute values from different policies.

Entitlements parameters

Each entitlement in a provisioning policy is also responsible for determining how access to a particular service is configured by defining parameters for a service. For example, you can determine if a user is a member of a group on a particular resource, where their home directory is created, and the level of access privilege. The following parameter types are valid:

- ▶ Constant value
- ▶ Null
- ▶ JavaScript
- ▶ Regular expression

The provisioning parameters in an entitlement can be defined statically or dynamically.

Parameters are defined statically by selecting the constant parameter type and specifying literal values, such as strings or integers. For example, an attribute can be defined as Domain Users or Power Users.

A parameter value that is defined dynamically can be based on a JavaScript function that can retrieve data from the Tivoli Identity Manager directory server. A range of values can be defined using a regular expression.

Parameters can also be specified as null, indicating that the parameter does not have a value. This situation is equivalent to having a JavaScript parameter type with a value of return null.

Provisioning parameters for single-valued attributes can be based only on JavaScript code or a constant. The provisioning parameters of a multi-valued attribute can use a constant, JavaScript code, or regular expression for their values. However, a regular expression can be used only for provisioning parameter enforcement of the Allowed or Excluded type.

Policy enforcement

Provisioning policies are very important to support security compliance. Tivoli Identity Manager evaluates all account and access requests based on the

provisioning policy to identify accounts and access that are not authorized and take appropriate actions to handle non-compliant account and access. Based on the enforcement configuration on the service, Tivoli Identity Manager can either:

- ▶ Mark the account or access as non-compliant
- ▶ Suspend the account
- ▶ Alert administrator to revoke disallowed privilege, or
- ▶ Automatically correct the account or access and make it compliant.

The advanced provisioning parameter list also contains enforcement fields that enable you to specify an *enforcement rule* for the attributes. Enforcement rules determine which attributes are required for an account and which values are valid for the attribute. Attribute enforcement is active only if at least one provisioning parameter is defined. If no parameters are defined, all attributes are implicitly allowed. An enforcement rule can have one of the following values:

- ▶ Default
- ▶ Mandatory
- ▶ Optional
- ▶ Excluded

A *null* provisioning parameter value has a special meaning in Tivoli Identity Manager:

- ▶ A null mandatory parameter value means that all values on the corresponding attribute of a new or existing account are disallowed. Any existing attribute values will be automatically removed.
- ▶ A null default for optional parameter value means that all new values or changes to existing values on the corresponding attribute of a new or existing account are disallowed, but currently set values will not be removed automatically. Any currently set value can be removed manually.
- ▶ A null excluded parameter means that all attribute values are allowed on the corresponding attribute of a new or existing account.

Note: Each service targeted by a provisioning policy can also have a workflow attached. Workflows are used to introduce additional processing before access is provisioned. Workflows can initiate an approval process for a particular service or set of services. We present more information in 4.6, “Workflows” on page 128.

Global policy enforcement is the manner in which the Tivoli Identity Manager system globally allows or disallows accounts that violate provisioning policies. When a policy enforcement action is global, the policy enforcement for any service is defined by the default configuration setting. However, if a service has a

specific policy enforcement setting, that setting takes precedence over the global enforcement setting.

Provisioning policy join directive

Provisioning policy *join directives* define how Tivoli Identity Manager manages attributes when provisioning policies conflict. Provisioning policy join directives take affect when there are multiple provisioning policies defined for the same users (or groups of users) on the same target service, service instance, or service type. Unlike the configuration policy itself, the join directive configuration and customization is defined under a separate part of the administration interface, using **Configure System** → **Configure Policy Join Behaviors**.

The entitlement target type also plays a role in how policy join directives resolve which entitlement is granted when conflicts arise between policies. When two or more policies grant similar entitlements, the more specific entitlement takes precedence. For example, if one provisioning policy includes an entitlement defined to grant access to a type of service (that is, AIX named AIX), and the second policy includes an entitlement defined to grant access to a specific instance of that service (that is, AIX105), the more specific entitlement takes precedence.

Tivoli Identity Manager provides several types of join directives.

- ▶ Union (the default for Multivalued string or number type of attributes)
- ▶ Intersection
- ▶ Append
- ▶ And
- ▶ Or (the default for Single-valued boolean string type of attribute)
- ▶ Highest (the default for Single-valued integer type of attribute)
- ▶ Lowest
- ▶ Average
- ▶ Bitwise_Or (the default for Singled-valued bitstring type of attribute)
- ▶ Bitwise_And
- ▶ Precedence_Sequence
- ▶ Priority (the default for Single-valued string type of attribute)

The provisioning policy with the lowest priority number takes precedence over a similar policy that grants the same entitlement with a higher priority number. If conflicting policies have the same priority, the first policy found by the system is used.

In addition, Custom join directives can be defined (using Java) to change the built-in join logic completely.

For a better understanding of priorities and join directives, see the examples provided in the IBM Tivoli Identity Manager Information Center:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/ref/ref_ic_policy_joindir_examples.htm

4.5.4 Service selection policy

A *service selection policy* is another way to specify a target type in a provisioning policy entitlement. Service selection policies extend the ability of provisioning policies by providing the ability to provision accounts based on attributes contained in a user's profile. In order for a service selection policy to be enforced, it must be the target of a provisioning policy.

The service selection policy then:

- ▶ Identifies the service type to target.
- ▶ Creates the account based on the JavaScript definition contained in the service selection policy.

The service selection policy can be located in the same container (business unit) as the provisioning policy or in a container located above the provisioning policy's container. The scope of a service selection policy determines which provisioning policies can target it. Service selection policies with single scope can only be targeted by provisioning policies at the same level in the organization tree as the service selection policy. Service selection policies with sub-tree scope can be targeted by provisioning policies at the same level or below the service selection policy.

Service selection policies are evaluated whenever a user is added to an organizational role (static or dynamic) that is a member of a provisioning policy that targets the service selection policy, when a user's attributes are modified, or when the policy itself is modified. If, as a result of the policy's evaluation, a user's account must be moved to a different service instance than the one the user is currently using, the system creates a new account for the user on the new service instance and completes one of the following actions based on the service instance's policy enforcement setting:

- ▶ Suspends the existing user account on the old service instance.
- ▶ Deletes the existing user account on the old service instance.
- ▶ Sends a work item to alert the recipient that the existing user account on the old service instance needs to be deleted.
- ▶ Marks the account on the old service instance as disallowed.

Keep in mind that a service selection policy is not in use until a provisioning policy targets it. Also, a service selection policy can be deleted only when no provisioning policy references it.

For an example of a service selection policy, refer to IBM Tivoli Identity Manager Information Center:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/ref/ref_ic_admin_servselpolicy.htm

4.5.5 Adoption policy

We discuss the adoption policy in “Adoption policy” on page 112.

4.5.6 Recertification policy

The recertification process validates that each user account is still required for a valid business purpose. Tivoli Identity Manager can simplify and automate this process using a *recertification policy*.

A recertification policy defines how frequently users must certify their need for account access. Additionally, using the workflow engine, the policy defines the operation that occurs if the recipient declines or does not respond to the recertification request.

Recertification policies target either accounts or accesses, but accounts and accesses cannot be mixed as a target.

All accounts on services are eligible for recertification, including the Tivoli Identity Manager Tivoli Identity Manager Service accounts and manual services. A service can be a member of only one recertification policy, and one or more services can be selected. If a service owner is creating the policy, only the services owned by the service owner are eligible for selection, based on default ACIs.

All access entitlements are eligible for recertification. An access entitlement can be a member of only one recertification policy, and one or more access entitlements can be selected. If a service owner is creating the policy, only the access entitlements available on the services owned by the service owner are eligible for selection.

Tivoli Identity Manager does not select accounts for recertification in the following cases:

- ▶ Tivoli Identity Manager *Manager* account (the actual name of the account and service inside Tivoli Identity Manager) to avoid deleting or suspending of the account accidentally.
- ▶ Orphan accounts, because they do not have owners.
- ▶ If the rejection action is to suspend, the recertification policy does not find accounts that are already suspended, because the rejection of the recertification will cause the accounts to be re-suspended. If the reject action deletes the accounts and the account was already suspended, it will be selected for recertification, because further action such as deleting the suspended account can be taken.

The recertification approval activity does not necessarily go to the account owner. The participant can be specified using the “Who approves recertification” field in the policy tab. Approvers can be the account owner, administrator, service owner, manager, user, organizational role, group, or access owner for access entitlements.

A recertification policy defines the content of an e-mail notification to participants and the interval that triggers a request for recertification. The e-mail notification alerts you to recertify a need to use an account or access. The action to be taken when the user ignores the request is specified using Timeout Action, which is set to Approve by default. The control can be set to reject also.

Because the recertification process can cause large volumes of e-mails and workflow pending requests, consider splitting accounts into different time slots for recertification schedule and invoke notification e-mails during the low production load hours.

4.5.7 Account defaults

Account defaults define default values for an account during new account creation. The default can be defined at the service-type level that applies to all services of that type, or at the service level, which only applies to the service. The service-type default can be overridden by defining an account default at the service level. Subsequent changes (including removals) to the account defaults on the service-type are not reflected to the pre-existing Services.

Account defaults can be hard coded values or can have some logic written in JavaScript.

An account default has a functionality that is also built into the provisioning policy. However, account defaults should be used to set up default account values for

non security-sensitive attributes, and provisioning policies should be used to set up account attribute constraints for security compliance. Good practice is also to avoid overlapping of the same attributes between the two. Account defaults can be overridden by the provisioning policy *entitlement default attributes*.

You can optionally choose to prevent new account defaults from being added to a service and prevent existing account defaults from being changed or removed on the service. The default value allows new accounts defaults to be defined on the service and to allow existing account defaults to be changed and removed on the service.

4.6 Workflows

A workflow defines a sequence of steps that represents the *business process*.

Tivoli Identity Manager has the following types of workflows:

- ▶ Account request workflow²
- ▶ Access request workflow
- ▶ Operation workflow

4.6.1 Account request workflow

An *account request and access request workflows*, in the most basic form, are a description of the approval process for a provisioning process. Both workflows are managed in the same part of the Administration interface—the **Manage Workflows**.

The account request workflow is invoked during account provisioning requests, including adding and modifying an account, made by a Tivoli Identity Manager user or made during account auto provisioning. An account request workflow can also be invoked during an access request if there is no access request workflow defined.

Thus, account request workflows are assigned to a provisioning policy. If there are multiple provisioning policies that apply to the same user for the same service target, and there are different account request workflows in each provisioning policy, the account request workflow that is invoked for the user is determined based on the priority of the provisioning policy. The workflow runs before the policy grants the entitlement. If the workflow returns a result of approved, the policy grants the entitlement. If the workflow has a result of rejected, the entitlement is not granted.

² This workflow was known as *entitlement workflow* in Tivoli Identity Manager Version 4.6.

An account request workflow is invoked during the following events:

- ▶ New account requests from a Tivoli Identity Manager user.
- ▶ Account modify requests from a Tivoli Identity Manager user.
- ▶ Automatic account provisioning from the Tivoli Identity Manager system when automatic entitlement is defined in the provisioning policy.
- ▶ New access requests from a Tivoli Identity Manager user, and there is no access request workflow defined for that access.
- ▶ Policy enforcement from a Tivoli Identity Manager system. If an account modify operation occurs because of a policy enforcement action, the account request workflow for an account is executed only if the following property in the `enRole.properties` file is set to `false` (default value is `true`):

```
enrole.workflow.skipfornoncompliantaccount=false
```

4.6.2 Access request workflow

This is a new type of workflow introduced in Tivoli Identity Manager v5.0.

The access request workflow can specify the steps and approvals that authorize access to resources in a request. The access request workflow is only invoked if a Tivoli Identity Manager user requests an access. It is not invoked during an account request, even if the access request workflow gives the user the same access by assigning the account to a specific group.

Use the *Define an Access* task for a service group to assign an access request workflow to a specific access.

4.6.3 Operation workflow

Managed objects in Tivoli Identity Manager are called *entities*. Categories or classes of managed objects are called *entity types*, and those are *accounts*, *persons*, or *business partner persons*.

An operation workflow defines the business logic for managing entity types and entities.

When an administrator adds, removes, or modifies one of these entities using the Tivoli Identity Manager administration user interface or the APIs, the operation workflows are used to execute the request. You can customize the operation workflows, modifying both the flow and specific activities in order to implement your specific business processes. You can also extend Tivoli Identity Manager capabilities by creating new operations and using the APIs, life cycle rules,

operation workflow activities, or custom workflow extensions to call those new operations.

Operation workflows are defined at global as well as lower levels in the system. The available operations depend on which operation level is selected. Opposite to account and access request workflows, operation workflows are defined in a different part of the administration interface using **Configure System** → **Manage Operations**.

You can define the operation workflows on the following operation levels:

- Global** Applies to all entities and entity types. A global operation is always a static operation that does not require a target entity.
- Entity type** Applies to all entities of that specific type, such as an Account or Person entity. For static operations, an entity type defines the namespace of the operation; for a non static operation, an entity type defines the type of the target entity. An operation at the entity type level does not overwrite an operation at the global level.
- Entity** Overrides the operations that are defined at the entity type level.

Operations are either static or non-static, which relates to the attributes of the target of the operation that relates to how the target of the operation is provided to the operation. Operation workflows support input parameters, but not output parameters.

Life cycle management

When an entity is created in Tivoli Identity Manager, the entity type provides a set of default characteristics and operations. Tivoli Identity Manager provides a set of system-defined entity type operations that you can use to extend and create user-defined entity operations, which override those entity type operations.

Defining operations for entities and entity types are the major ways with which you can readily implement your specific business process requirements.

To customize the life cycle management of operations for entities and entity types, you use operation workflows.

Life cycle management was first introduced in Tivoli Identity Manager V4.5 to customize the operations involved in account and person management, including the definition of entirely new (custom) life cycle operations for account and person entities.

For example, it is possible to create a custom business approval processes for each of the add, modify, suspend, restore, and delete operations of an account,

or to provide custom logic for creation of a person and its attributes, such as deriving some of the attributes based on other pre-populated attributes.

Tivoli Identity Manager life cycle rules can be used to automate the often large number of manual tasks that administrators must perform due to changes in the environment. These changes include common reoccurring events such as account inactivity, password expiration, or contract expiration, which are driven by business policies. Life cycle rules can also eliminate the potential of some policies to go unenforced.

Each rule can be defined in one of three ways:

- ▶ Global
- ▶ Associated with an entity type
- ▶ Associated with an entity

Life cycle rules are similar to life cycle operations, and the definition of life cycle rules are available from the same user interface as the operations themselves.

Establishing life cycles enables Tivoli Identity Manager administrators to define events that can be triggered based on a time interval or based on time and matching criteria evaluated against an entity. The administrator can then associate life cycle operations that is executed as a result of that event. All life cycle rules consist of two parts:

- ▶ The definition of an event that triggers the rule
- ▶ The identification of the life cycle operation that executes the actions specified in the rule

4.6.4 Workflow elements

Any type of workflow is made up of one or more of the following elements:

- ▶ Processes

Processes define the activities and flow between activities that are needed to execute a business process. Processes include *activities*, *transitions*, *input/output parameters*, and *relevant data*.

- ▶ Activities

Activities represent the business logic for a specific task in a workflow process. An activity is represented in a workflow as a *node*. Tivoli Identity Manager supports the following types of nodes:

- Start and End

The start node defines the beginning of a workflow, and the end node defines the end of a workflow. Both nodes are always included in a

workflow and cannot be deleted. These nodes each contain a JavaScript window that allows you to add JavaScript code that executes at the beginning or end of the workflow. Start nodes have transitions out only and end nodes have transitions in only.

- Approval

Use the approval node to add a request for approval when adding or modifying people, accounts, and access. The approver must be a Tivoli Identity Manager user, because the approver is required to log in to Tivoli Identity Manager to approve or reject the request. In entitlement workflows, use approval nodes to request authorization to continue with a provisioning request. In operation workflows, use an approval node as a switch to follow a specific workflow path. Approval text and labels can also be customized to allow approvals to be used for most Yes/No decision activities.

- Mail

Use the mail node to specify the recipient type and content to be e-mailed to a user in an e-mail notification. The content can be specified directly or copied from a template used by mail activities in other workflows.

- Request for information

Use the request for information (RFI) node within entitlement and operation workflows to solicit account or user-related information from a user with a Tivoli Identity Manager account. Within the RFI, you specify the attributes for which the participant is asked to provide values. The participant can edit only the attributes that you select. All other form attributes are read-only. The page displayed in the Activities to-do list matches the form that is specified using the form designer. Attributes listed as mandatory on the account form are mandatory also for the RFI.

You do not need to create ACI definitions for the fields to which the participant is asked to respond. You must select an entity type and entity for the RFI to be used to request information about attributes for a specific entity. After the entity type is selected, a list of attributes displays from which to select. The attributes selected display on the RFI page when the participant logs in and accesses the RFI activity list item.

- Operation

Use the operation node to invoke an existing operation from within a workflow.

- Loop

Use the loop node to execute one or more nodes in a loop. Keep in mind that nested loops are not supported. Nodes contained within the loop must not transition to any activities outside the loop. The loop node does not

specify the results of the nodes in the loop. You must check the status of nodes in the loop in a script following the loop if required.

Because an activity in the loop can execute multiple times (once for each loop iteration), the workflow engine keeps track of activities in a loop by giving them an index representing which iteration of the loop that the instance of the activity applies to. This index is stored in the activity object as a member variable called `index`. For activities that are not in a loop, this index is set to 0. For activities in a loop, this value will be $1-n$, where n represents the number of actual loop iterations that execute.

There are two types of loops:

Do while Evaluates the condition prior to executing. If the condition is true, the loop executes. Otherwise, it continues with the next node. This loop type is used when the condition is dependent on the workflow processes prior to the loop node. The process that is defined by the loop will not run if the condition is already met.

Do until Evaluates the condition after each execution of the loop nodes. The workflow completes the process that is defined in the loop before checking the loop condition. If the condition is true, the loop executes again. Otherwise, it continues with the next node after the loop. This loop type is used if the process defined by the loop must run at least once regardless of any previous activities.

– Extension activities

Use the extension node to invoke an application extension from within the workflow. The application extension is a preconfigured Java class for use in the workflow environment. Extensions can accept input parameters and return output parameters back to the workflow. Only extensions that are registered properly display in the extension window.

– Script

Use the script node to add logic to the workflow through the use of JavaScript code. The script node makes clear to anyone viewing the workflow that scripting is present in the workflow. JavaScript code is used within workflows to define and retrieve parameters and attribute values dynamically and to store and forward these values as variables for use by logic or code within a single workflow activity. You can extend the JavaScript code by defining custom JavaScript objects through a Java extension.

- Workorder

Use the workorder node to send e-mail to a Tivoli Identity Manager user, either to request some type of manual activity or as a simple notification. The work order activity supports two execution modes:

- The *send mode* completes the activity when the work order request messages are successfully sent to the mail server for forwarding to participants.
- The *send and wait for completion mode* sends the e-mail and then waits for notification of the completion of a manual activity.

- Subprocess

This node is available only for account request and access request workflows. A subprocess activity cannot be included in an operation workflow. A subprocess can use any predefined account request or access request workflow of the same service type (or global workflows); however, the workflow must be located within the same organization.

- ▶ Transitions

Transitions represent a flow between two activities. All design elements are connected with transition lines. When a workflow process is executed, the flow from one activity to another activity is controlled by the conditional logic (JavaScript coding) in the transitions and the activity configuration information. You can define both serial and parallel flows with Tivoli Identity Manager.

- ▶ Input and output parameters

Input and output parameters define the data that is passed into and returned from a workflow process or some activity defined in the workflow. Some workflow processes in Tivoli Identity Manager might restrict the customization of input and output parameters.

- ▶ Relevant data

Relevant data defines global variable data for workflow processes. You can use this variable data to pass data from one activity to the next by associating it with activity parameters. Output parameters resulting from an activity are stored as a relevant data item. They are then passed from relevant data and become the input parameter for another activity.

Transitions can also access and use relevant data in their conditional logic.

- ▶ Activity participants

Activity participants are Tivoli Identity Manager users who have been assigned to interact with activities in a workflow process, such as approvals, mail, requests for information, and work orders. An activity participant can be a specific user with a Tivoli Identity Manager account or assigned to a

particular organizational role, or a user with a specific relationship, such as a supervisor or services owner.

Some workflow activities might restrict the list of available participants, whereas some activities (such as mail or work order activities) can be configured to not require the participant to be a Tivoli Identity Manager user at all.

- ▶ JavaScript

Many of the workflow elements, such as transitions and script elements, integrate the JavaScript scripting language to enable customization of workflow processes. In addition to the standard JavaScript extensions, Tivoli Identity Manager provides JavaScript extensions that you can use to access processes, activities, relevant data, and participants.

A process name uniquely identifies each element within a workflow. Therefore, each workflow element can be thought of as its own process within Tivoli Identity Manager. The process name is required when creating each element in the workflow. The easiest way to view an element's process name is to double-click the element to view the properties.

Optionally, a workflow can be added to each entitlement. Adding a workflow requires that at least one approval occurs before the service or access request is granted. Workflows are attached to provisioning policies by clicking the Entitlement tab and choosing the predefined workflow from a list within the Process Definition field of the Entitlement panel.

Workflows frequently contain activities that require manual intervention or input from a person before they can complete. When an activity requires an action before it can continue, a *To Do item* is assigned to the user, and the To Do item displays in the user's To Do list, which can be viewed the next time the user logs in to Tivoli Identity Manager. This To Do item might consist of a user approving or rejecting a request, providing the activity with information for it to process, or completing a manual task outside of the system. When a user completes the To Do item, the activity can then proceed. This user is called a *workflow participant*. Workflow participants must have a Tivoli Identity Manager account in order to respond to approval requests and RFIs. Workflow participants that respond to work order requests, which must be completed before the workflow can continue, must have a valid e-mail address or a Tivoli Identity Manager account.

For each activity that requires a participant, you can also specify an *escalation participant*. When you specify an escalation participant, you define a time limit that is used to determine when the request should be escalated. If the original participant cannot be resolved or does not respond within the specified time, the escalation participant is notified. If for any reason the participant and the

escalation participant cannot be resolved, the System Administrator group becomes the workflow participant.

4.6.5 Workflow notification properties

You can configure some workflow properties to apply globally to workflows in Tivoli Identity Manager. You can configure Tivoli Identity Manager with a *default escalation period* that is used to determine when work items resulting from workflow activities are escalated. You can also customize activity notification message templates to send notifications.

All workflow activities are escalated when the escalation period expires. The default escalation period serves as the initial value for newly defined workflow activities. To override the default escalation period, configure the escalation period for a specific activity contained in a workflow.

You can also configure Tivoli Identity Manager to send activity notifications and to-do list item reminders through e-mail to workflow participants after a configured amount of time. The *Reminder Interval field* specifies the time in days. The value cannot be less than the time interval for the escalation limit.

Tivoli Identity Manager sends e-mail notifications for specific type of account requests and for specific events in the workflow system. We discuss the e-mail notifications in 4.12, “Post office” on page 147.

4.7 Tivoli Identity Manager groups

Tivoli Identity Manager *groups* are collection of people with Tivoli Identity Manager accounts. You use these groups to administer a type of user’s access to Tivoli Identity Manager. Access rights and views within Tivoli Identity Manager must be assigned to groups rather than individuals. Therefore, the Tivoli Identity Manager group to which your Tivoli Identity Manager account is assigned determines what you are allowed to do within Tivoli Identity Manager.

Note: There is no requirement that every person have a Tivoli Identity Manager account. Only those people that need to access the system need accounts. If your organization is implementing self-service, each user requires a Tivoli Identity Manager account.

By default, Tivoli Identity Manager provides the following types of groups:

- ▶ System administrator

The person who belongs to this group is responsible for a variety of Tivoli Identity Manager setup and administration activities such as provisioning people, adding services, defining access entitlements, and setting permissions for system users.

- ▶ Service owner

The person who belongs to this group is responsible for enabling users to perform tasks that are associated with services and access entitlements.

- ▶ Help desk

The person who belongs to this group is responsible for assisting users with common user and account management tasks, such as locked accounts and passwords.

- ▶ Manager

The person who belongs to this group is responsible for users who report to them.

- ▶ Auditor

The person who belongs to this group is responsible for auditing the system by creating reports.

You might create additional groups, called *custom groups*, to specify a set of data and functions for which a collection of users have responsibility.

Tivoli Identity Manager provides one default account automatically (*itim manager* with the initial password *secret*), which is preconfigured with the system administration role. System administrators have access to any part of the system and are not governed by ACIs.

By default, a new user is created as a system user (non-administrative user). A *system user* is a common user of resources whose identity is managed by Tivoli Identity Manager. By default, a system user does not belong to any Tivoli Identity Manager group, in which case the user does not have access to the administrative console but can access the self-service application.

Tivoli Identity Manager groups cannot be filter based. They contain only a static list of members. A user can be a member of more than one group and can obtain membership in a group either manually using the administration console or automatically by reference in following cases:

- ▶ If the user is assigned as a manager of a person, that user is added to the Manager group by default.
- ▶ If the user is an owner of a service instance, that user is added to the Service Owner group by default.

As a prerequisite for this implicit group membership, the security property *Automatically populate Tivoli Identity Manager group* must be enabled (it is disabled by default).

When a user is removed as a manager of another user or the owner of a service loses owner privilege, that user is not removed automatically from the respective groups. In that case, Views and some ACIs still apply until they are removed from the groups.

Manual action needs to be performed to remove a user from any group, including Service Owner and Manager. The only exception is if a person record is deleted from Tivoli Identity Manager. In that case, Tivoli Identity Manager removes user memberships automatically from the default or custom groups.

Groups are used in conjunction with default ACIs and Views to create a default administration and security functionality of the Tivoli Identity Manager system.

4.8 Access control item

An *access control item* (ACI) is the mechanism by which Tivoli Identity Manager governs access rights to its system users. Tivoli Identity Manager comes with large sets of predefined ACIs but also provides the ability to modify existing ACIs and to create new, custom ACIs.

Let us take a look at some principles for designing ACIs:

- ▶ Users can set their own passwords, view their own personal information, see their own accounts and request new accounts, or view defined accesses (self-service).
- ▶ A manager can manage subordinates (relationship).
- ▶ Service owners can manage the services that they own and the accounts that are hosted on that service (relationship).

- ▶ Service owners can manage service groups or access entitlements for the services that they own.
- ▶ Access owners can search services for the accesses that they own.
- ▶ A help desk user can manage users and their non-admin accounts (root/administrator, by default) and can reset or change user passwords as appropriate.
- ▶ An auditor (group) can perform the following tasks:
 - See (run) all reports for all the data and events in the system.
 - See the entire audit trail and view all data in the system, but not be authorized to change any data beyond own basic data as a user.

After creating a new access control item or changing an existing access control item, run a data synchronization to ensure that other Tivoli Identity Manager processes, such as the reporting engine, use the new or changed access control item.

An access control item defines:

- ▶ The entity types to which the access control item applies.
- ▶ Operations that users might perform on entity types.
- ▶ Attributes of the entity types that users might read or write.
- ▶ The set of users, organized in groups, who are governed by the access control item.

An access control item can focus on the following categories of entity types:

- ▶ Account
- ▶ Account Default Template
- ▶ Admin Domain, which identifies a subsidiary part of an organization as a separate entity with its own policies, services, and access control items, including an administrator whose actions and views are restricted to that domain.
- ▶ Business Partner Organization, which identifies a business partner organization, which is typically a company outside your organization that has an affiliation, such as a supplier, customer, or contractor.
- ▶ Business Partner Person, which represents an employee of an outside entity with which your organization is affiliated, such as a supplier or customer.
- ▶ Dynamic Organizational Role
- ▶ Group
- ▶ Service Group
- ▶ Location

- ▶ Organizational Unit
- ▶ Person
- ▶ Identity Policy
- ▶ Password Policy
- ▶ Provisioning Policy
- ▶ Recertification Policy
- ▶ Service Selection Policy
- ▶ Report
- ▶ Service
- ▶ Static Organizational Role
- ▶ Workflow Design, which defines who can create or modify account and access entitlement workflows.

When creating an ACI, you can choose one of the following ACI *scopes*:

- ▶ Single scope: The ACI manages rights at the same level it was created.
- ▶ Sub-tree scope: The ACI manages rights at the same level it was created, as well as all levels below it in the tree.

Tivoli Identity Manager operates on the basis of the following permissions:

- ▶ Grant: The permission is explicitly granted.
- ▶ Denial: The permission is explicitly denied.
- ▶ None: The permission is neither granted nor denied (denial by assumption).

There are two permission types controlled by ACIs:

- ▶ Attribute permissions: Grants access to read or write to data fields, which contain information (attributes) about a target (entity or process). For example, if you select person as your target, Tivoli Identity Manager provides a list of attributes to which you can grant access, such as user ID, home address, and telephone number.
- ▶ Operation permissions: Grants access to specific operations for the selected target. For example, if you select organizational unit as your target, Tivoli Identity Manager provides a list of operations to which you can grant access, such as remove, search, add, and modify.

4.8.1 Conflicts between multiple ACIs and Tivoli Identity Manager groups

In some situations, you might have multiple ACIs applying to the same person for the same task. A person who has memberships to more than one Tivoli Identity Manager group or multiple ACIs linked to the same Tivoli Identity Manager group can cause this conflict to happen.

In this case, it is possible that a person is granted rights by one ACI that are denied by another ACI. In the case where two or more ACIs conflict, the following actions occurs:

- ▶ Explicit deny overrides explicit grant.
- ▶ Explicit grant overrides implicit deny, also known as *none* (the default).

In the case of multiple Tivoli Identity Manager groups, a person's access is enabled based on the widest privilege assigned to any of the person's assigned Tivoli Identity Manager groups. Any access explicitly denied to a person in one group is denied to the person in all groups. Because positive denials override all other choices, it is very important to use deny sparingly.

4.9 Views

A *view* is a set of tasks that a particular type of user can see, but not necessarily perform, on the graphical user interface. On both the self-service console and the administrative console, you can specify the view (set of tasks) that a user sees. Tivoli Identity Manager provides default views of the tasks that are available for each default group:

- ▶ End User View
- ▶ Manager View
- ▶ Help Desk View
- ▶ Service Owner View
- ▶ Auditor View

A special feature for the Service Owner view of the Admin Console is the *Service Owner Dashboard*. The home page displays "Service Connection Status" for the services owned by this user. A summary status is provided.

If you give a user or group a view, you do not give permissions to the user or group to perform the functions within that task. You must also define access control items to give the user or group the necessary permissions for the task.

If a user is a member of multiple groups, the user inherits view of all of the tasks that are provided to both groups, even if one of the groups does not grant access to the task.

An administrator can configure custom views by going to **Set System Security** → **Manage Views** → **Configure View**. Each view definition allows specification of the Self Service Console view and the Admin Console view.³

Thus, you need to coordinate the views that users see when they have memberships in multiple groups. One group to which they belong might permit a task, and another group might not. If a task is permitted in any view, that permission takes precedence. For example, if a task is permitted in the view that one group has, a user in that group can use the task, even if the user is also member of a second group with a view that excludes the same task.

4.10 Auditing

Tivoli Identity Manager Server provides auditing for any actions taken by a Tivoli Identity Manager user that changes a business object or the configuration of the system. The following list describes all auditing events:

- ▶ ACI Management (Add, Add Authorization Owner, Delete, Delete Authorization Owner, Modify)
- ▶ Account Management (Add, Adopt, Change Password, Delete, Modify, Orphan, Password Pickup, Restore, Suspend, Synchronize Password)
- ▶ Access Management (Add, Remove)
- ▶ Access Configuration (Add, Remove, Modify)
- ▶ Authentication (Authenticate Tivoli Identity Manager user)
- ▶ Container Management (Add, Delete, Modify)
- ▶ Delegate Authority (Add, Delete, Modify)
- ▶ Entitlement Workflow Management (Add, Delete, Modify)
- ▶ Entity Operation Management (Add, Delete, Modify)
- ▶ Tivoli Identity Manager Configuration (Add, Delete, Enforce, Install Profile, Modify, Uninstall Profile)
- ▶ Group Management (Add, Add Member, Delete, Modify, Remove Member)
- ▶ Migration (Agent Profile Install, Start Export, Start Import, Stop Export, Stop Import)

³ For more information, see 4.16, “User interface customization” on page 160.

- ▶ Role Management (Add, Add Member, Delete, Modify, Remove Member)
- ▶ Person Management (Add, Delete, Modify, Restore, Self Register, Suspend, Transfer)
- ▶ Policy Management (Add, Commit Draft, Delete, Enforce Entire Policy, Modify, Save as Draft, Add Account Template, Change Account Template, Remove Account Template)
- ▶ Reconciliation (Run Recon, Set Recon Unit, Set Service Recon Parameters)
- ▶ Runtime Events (Start Tivoli Identity Manager, Stop Tivoli Identity Manager)
- ▶ Self Password Change (Change Password, Reset Password)
- ▶ Service Management (Add, Add Adoption Rule, Delete, Delete Adoption Rule, Modify, Modify Adoption Rule)
- ▶ Service Policy Enforcement (Correct Non-Compliant, Mark Non-Compliant, Suspend Non-Compliant, Use Global Setting, Use Workflow For Non-Compliant)

Internally, audit event records are stored in Tivoli Identity Manager database tables.

Auditing is one of many features in Tivoli Identity Manager that does not have an appropriate user configuration interface. Configuration of auditing (turning on or off) is done in the `enRoleAuditing.properties` configuration file by setting following line to a true or false value:

```
itim.auditing=true
```

This file also contains entries for all auditable events that can be turned off or on. For more details about the configuration of auditing, see the IBM Tivoli Identity Manager Information Center:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

Audited information includes the identity of the user taking the action, the time the action was taken, the type of action taken, and the data affected by the action.

You can use a summary of audited information to generate a report.

4.11 Reporting

The reporting system enables users to generate reports from the Tivoli Identity Manager database based on selected criteria. Reports organize system activity

information according to specific criteria and display the results in a format that users can view or print.

Report data is staged through a data synchronization process, which gathers data from the Tivoli Identity Manager directory information store and prepares it for the reporting engine. Data synchronization can be run on demand, or it can be scheduled to occur regularly.

Tivoli Identity Manager provides two types of reports:

- ▶ Standard reports that are supplied with Tivoli Identity Manager
- ▶ Custom reports

All reports, including standard reports and custom reports, are generated using *report templates*. A report template defines the layout of a report and the filter criteria that determines the contents of the report. When you select a report to run, you are selecting the report template used to generate the report.

All reports are generated in standard PDF format and need Adobe® Acrobat® Reader for reading. Additionally, reports can be generated in comma-separated value (CSV) format, which gives additional flexibility for report data to be imported in additional tools for analysis.

Tivoli Identity Manager provides a large set of standard report templates that are designed to help you manage system resources and monitor the status of various activities and accounts. You can keep the standard report templates in their original form to generate reports, examine them to determine how to design custom report templates, or modify the standard report templates to meet the needs of your organization. You modify standard report templates using the **Design Report** task in the console.

Custom reports are generated using report templates that are designed using either the built-in report designer or a third-party report designer, such as the Crystal Reports report designer. The Crystal Reports designer is run separately, outside of the Tivoli Identity Manager console. If you do not configure the Crystal Reports designer to design, view, and generate reports, only the built-in report designer is made available in the console.

When you, as an administrator, create a custom report, you must also create report ACIs and entity ACIs manually for that custom report. These ACIs allow users who are not administrators, such as auditors, to run the custom report and to view data in the custom report.

By default, users or members of the Help Desk group cannot access reports. Besides administrators, there are default report ACIs for the Manager, Service Owner, and Auditor groups.

By default, the following categories of reports are available:

► **Requests**

- **Account Operations:** A report that lists all account requests. Allows filtering by account operation, service and other fields.
- **Account Operations Performed by an Individual:** A report that lists account requests made by a specific user. Allows filtering by the user who made the request in addition to other fields.
- **Approvals and Rejections:** A report that lists request approval activities that were approved or rejected. Allows filtering by activity approver, service and other fields.
- **Operation Report:** A report that lists all operations submitted in the system. Allows filtering by requestee, operations, and requests start and end date.
- **Pending Approvals:** A report that lists the request activities submitted but not yet approved. Allows filtering by service, activity status and other fields.
- **Rejected Report:** A report that lists all rejected requests. Allows filtering by requestee and request start and end date.
- **User Report:** A report that lists all requests, shows the set of operations that were requested, who the operations were requested for, and who requested them. Allows filtering by requestor, requestee, request start and end date

► **User and Accounts**

- **Account Report:** A report that lists services that user can select from to generate accounts report for a business unit. Allows filtering by service and business unit.
- **Accounts/Access Pending Recertification Report:** A report that lists all pending recertification activities. Allows filtering by account/access owner, service type, and service.
- **Individual Access:** A report that lists all user accesses and their owners. Allows filtering by a user that owns accesses, business unit of the user, access entitlement defined in the system, and service where access is supported.
- **Individual Accounts:** A report that lists the accounts and their owners. Allows filtering by user.
- **Individual Accounts by Role associated with Provisioning Policy:** A report that lists accounts owned by users of a specific role which is a member of provisioning policy. Allows for filtering by role and business unit.

- **Recertification Change History Report:** A report that lists recertification history of accounts and user accesses. Allows filtering by account/access owner, recertification response, start and end dates, and other fields.
- **Suspended Individuals:** A report that lists all individuals that have been suspended. Allows filtering by date
- ▶ **Services**
 - **Reconciliation Statistics:** A report that shows the activities that happened during the last completed reconciliation of a service, regardless of when the report data was synchronized. Remote services provide reconciliation statistics during a reconciliation. This report contains data from the last service reconciliation. Data synchronization is not a report prerequisite. Allows filtering by service.
 - **Services:** A report that lists services currently defined in the system. Allows filtering by service type, service, owner, and business unit.
 - **Summary of Accounts on Service:** A report that lists the accounts on a particular service. Allows filtering by service and account status.
- ▶ **Audit and Security**
 - **Access Control Information (ACIs):** A report that lists all access control items in the system. Allows filtering by access control item name, protection category, object type, scope, and business unit.
 - **Access Report:** A report that lists all access entitlements defined in the system. Allows filtering by access type, access entitlement, service type, service and administration owner of an access entitlement.
 - **Audit Events:** A report that lists all audit events. Allows filtering by audit event category, action, initiator, start date, and end date.
 - **Dormant Accounts:** A report that lists the accounts that have not been used recently. Reconciliation needs to be performed on a service. Allows filtering by service and dormant period.
 - **Entitlements Granted to an Individual:** A report that lists all users with the provisioning policies that they have been entitled. Allows filtering by user.
 - **Non-Compliant Accounts:** A report that lists all accounts that are non-compliant. Allows filtering by service and non-compliance reason.
 - **Orphan Accounts:** A report that lists all accounts that do not have an owner. Allows filtering by service and account status.
 - **Policies:** A report that lists target and memberships of the provisioning policies in the system. Allows filtering by name of the policy.

- **Policies Governing a Role:** A report that lists all provisioning policies for a given organization role. Allows filtering by role name.
 - **Recertification Policies Report:** A report that lists all recertification policies. Allows filtering by policy target type, service type, service, access type, and access.
 - **Suspended Accounts:** A report that lists the accounts that have been suspended. Allows filtering by user, account, service and date.
- ▶ **Custom**

The administrator must define a schema for each custom report template that is created, (including designer reports and Crystal Reports) because entities and attributes are not included in custom reports. To create a report schema, you must run the Design Schema task in the console.

A *report schema* specifies which entities and attributes can be included in reports. Before an entity and its associated attributes can be specified as reporting criteria and included in custom report data, a report schema must be defined.

The schemes are installed for all of the standard reports during product installation. The administrator does not define schemes for standard reports.

By defining the schema, you select directory entities that will be staged as tables in the Tivoli Identity Manager database. Defining the schema involves mapping attributes. After mapping the entities and attributes, you must synchronize the data to make the data available for reporting. Only the entities and attributes for which you want to generate custom reports or Crystal reports should be mapped, because these mappings directly impact the performance of Tivoli Identity Manager. All of the data from the directory server is copied to the database each time a data synchronization is performed.

4.12 Post office

Tivoli Identity Manager uses an e-mail subsystem, based on the SMTP standard protocol, that can generate various e-mail notifications. Relying solely on an SMTP-based e-mail system has some limitations and can create problems and bottlenecks. For example, Tivoli Identity Manager might send a great number of identical or similar e-mails to people that are participants in workflow activities. An increased frequency and volume of e-mail notifications can also cause enterprise problems such as increased network traffic and increased load on mail servers.

To resolve these kinds of issues, Tivoli Identity Manager uses a *post office*. The post office provides a mechanism for reducing the number of e-mail notifications a user receives regarding similar tasks in Tivoli Identity Manager. The post office can be configured to collect similar notifications for a period of time and combine those into one notification that is then sent to a user.

Figure 4-2 shows configuration parameters for the post office. The figure shows that you can enable or disable the post office (by using the *Enable store forwarding* check box) and set the time interval (*Collection interval* field) that the post office uses to collect messages and aggregate them into single e-mail. You can also customize the e-mail template that is used to generate the aggregate message that is sent to the recipients.

The post office e-mail template can use dynamic content. Dynamic content includes dynamic content message tags, JavaScript code, and tags that replace variables with other values, or reference a property that allows translation with the use of a CustomLabels.properties file.

The following post office dynamic content custom tags can be used to get data:

- ▶ <POGetAllBodies/>
Returns a string containing the text body of each of the original notifications separated by a new line.
- ▶ <POGetAllSubjects/>
Returns all subjects from the notifications associated with the aggregate e-mail notification as a string that is separated by a new line.
- ▶ <POGetEmailAddress/>
Returns the e-mail address that is the destination for the aggregate e-mail notification as a string with no new line.
- ▶ <POGetNumOfEmails/>
Returns the number of e-mails that are associated with the aggregate e-mail notifications as a string with no new line.

Configure System > Post Office

To combine similar e-mail notifications into a single notification, type a notification interval that aggregates all e-mails. Then, type the subject, the plain text body, and dynamic XHTML body of the notification. To send a test message to an e-mail address, click Test. When your changes are complete, click OK.

Enable store forwarding

Collection interval
60

Aggregate Message

The Aggregate Message tab contains fields that are used to define a template that determines how the aggregate e-mail notification is displayed to a user.

Subject
<RE key="postoffice_subject"><PARM><POGetNumOfEmails /></PARM></RE>

Plaintext body
<RE key="postoffice_subject">
<PARM><POGetNumOfEmails /></PARM>
</RE>

<RE key="postoffice_subject_list" />
<POGetAllSubjects />

<RE key="postoffice_body_list" />
<POGetAllBodies />

XHTML body

OK Cancel Test

Figure 4-2 Post office configuration screen

If the post office is enabled, the message aggregation will not happen until the manual activities that generate notifications have the *Use Group E-mail Topic* option enabled. Only in that case, the post office intercepts e-mail notifications that the system generates for those manual activities and holds them for a specified interval.

When the collection interval expires and notifications are aggregated, if there is only one notification for a given Group E-mail Topic value and e-mail address, that message is sent in its original form (the post office e-mail template is not applied). Note that although the notification is sent in its original form, the notification is delayed until the post office collection interval expires.

If there are any errors while attempting to aggregate the individual e-mails, the messages are sent in their original form, and an error message is written to the log. Thus, notifications might be delayed in being sent, but the delay should not result in the loss of any notifications. The Test button located on the Post Office page is useful for troubleshooting template errors.

It is good practice to test and validate the post office e-mail aggregation template that you created before sending it to an activity participant. When performing the test, you must specify an e-mail address to receive the test message. The test, performs validation of the e-mail aggregation template, and if successful, a sample e-mail notification is sent to the e-mail address you specified. The e-mail message contains simulated system information, which is supplied by default in the properties file and is presented in the post office e-mail template that you created.

4.12.1 E-mail notifications templates

Tivoli Identity Manager sends e-mail notifications for specific type of account requests and for specific events in the workflow system. The notification can be enabled or disabled based on the request type or event type; and the notification template can be customized for each type of notification.

The following is a list of account requests in which an e-mail notification can be generated:

- ▶ New account
- ▶ New password
- ▶ Change account
- ▶ Deprovision account
- ▶ Suspend account
- ▶ Restore account

The following is a list of workflow system events in which an e-mail notification can be generated:

- ▶ Activity timeout
- ▶ Process timeout
- ▶ Process complete
- ▶ Approval work item
- ▶ Request for input work item
- ▶ Work order
- ▶ Compliance alert
- ▶ Work item reminder

Tivoli Identity Manager can also be configured to send activity notifications and to-do list item reminders through e-mail to workflow participants after a configured amount of time. Tivoli Identity Manager provides the ability to create default notifications for a type of activity in the form of templates. Notification templates provide a consistent notification style and content across manual activities and system activities such as adding accounts and changing passwords.

4.13 Configuring commonly used system properties

During the installation process, the Tivoli Identity Manager installation program runs the runConfig system configuration tool automatically to edit commonly used system properties for the Tivoli Identity Manager Server and also to configure WebSphere Application Server settings for the Tivoli Identity Manager application. The Tivoli Identity Manager installation program runs the system configuration tool for both a single-server and cluster configuration, which includes the deployment manager and the cluster members. You can run this tool manually to make changes to the system after the installation process:

```
ITIM_HOME\bin\runConfig install
```

In this command, ITIM_HOME is the root folder or directory of the Tivoli Identity Manager Server installation.

When you execute the command, a dialog box opens, as shown in Figure 4-3.

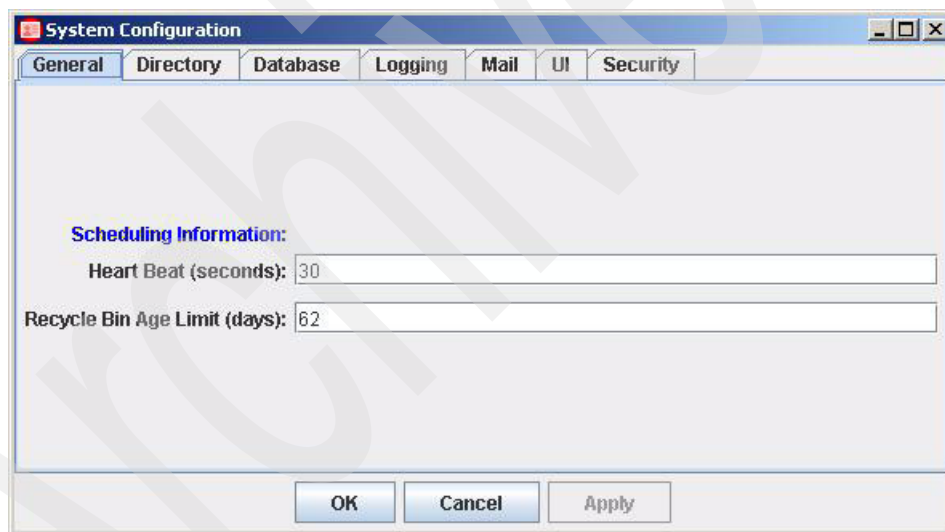


Figure 4-3 runConfig dialog box: Linux example

The dialog box includes the following configuration tabs:

- ▶ The General tab contains the general information about the Tivoli Identity Manager Server.
- ▶ The Directory tab contains directory connection information and LDAP connection pool information.

This tab also has a Test button to test the connection to the directory server. If you update any field on this tab, click **Test** to ensure that the connection works.

- ▶ The Database tab contains general database information and database pool information.

This tab also has a Test button to test the connection to the database. If you update any field on this tab, click **Test** to ensure that the connection works.

- ▶ The Logging tab enables you to set the level of tracing. It is possible to choose one of three values: MIN, MED, and MAX.
- ▶ The Mail tab contains mail notification and gateway parameters.
- ▶ The UI tab contains basic information to customize the Tivoli Identity Manager Server GUI.
- ▶ The Security tab contains information to manage database, LDAP, and application server user IDs and passwords that are stored in Tivoli Identity Manager properties files. This tab displays the encryption settings and application server user management preferences in the Tivoli Identity Manager Server.

For more information and details about the configuration fields, see *IBM Tivoli Identity Manager Server Version 5.0 Installation and Configuration Guide*, SC32-1562.

In addition to this tool, there are two other tools that the installation wizard executes in order to configure initially the directory server and database:

- ▶ IdapConfig
- ▶ DBConfig

Watch out! After the system is installed, do *not* use these tools to update any parameters because they will override all system parameters.

Running the **IdapConfig** command restores default values that Tivoli Identity Manager uses and *will cause loss of LDAP data*. If the Tivoli Identity Manager database tables have been previously set, running the **DBConfig** command drops all previously existing Tivoli Identity Manager tables. If you run this command after installation, ensure that the messaging engines under the service integration bus (itim_bus) have been stopped from the WebSphere Application Server administrative console before running **DBConfig**.

4.14 Modifying system properties manually

Instead of using the runConfig utility, you can modify system properties manually by editing the appropriate property file. System and supplemental property files are located on the Tivoli Identity Manager Server in the ITIM_HOME/data directory. These files contain all of the system and supplemental properties used by the server.

In addition, Tivoli Identity Manager uses a number of properties files to control the functionality of the program and to enable user customization of special features. Those properties files are also called Java-based properties files. Java properties files define the values of named resources that can specify program options such as database access information, environment settings, and special features and functionality. A properties file defines named resources using a property key and value pair format:

```
property-key-name = value
```

The *property-key-name* is an identifier for the resource. The *value* is the name of the actual Java object that provides the resource.

The enRole.properties file is one of the of major Tivoli Identity Manager system configuration files. It contains various configuration parameters that can be divided into the following areas:

- ▶ WebSphere Application Server properties
Define values that are specific to integrating Tivoli Identity Manager with the WebSphere Application Server.
- ▶ Application server properties
Define properties that are specific to the application server, such as a user-selected locale.

- ▶ Organization properties
Define the organization name that is used by the directory server.
- ▶ LDAP server properties
Define the properties that are used by used by the directory server in which Tivoli Identity Manager stores data.
- ▶ Search and LDAP control properties
Used to configure search strategy and LDAP control.
- ▶ Profile and schema cache properties
Define system cache performance.
- ▶ Messaging properties
Configure the internal communication between components of the Java Message Service (JMS) used by Tivoli Identity Manager.
- ▶ Scheduling properties
Used to configure the internal scheduler that runs calendar-based, scheduled events.
- ▶ Password transaction monitor properties
Used to check responses to password transactions and expire those transactions where the user has failed to respond within the allowed interval
- ▶ LDAP connection pool properties
Used to configure cache connection requests to the directory server.
- ▶ Passwords encryption properties
Used to configure password encryption.
- ▶ Challenge response encoding properties
Determine whether a response is encoded as case sensitive or insensitive.
- ▶ System listening port properties
Used to configure the listening port settings for the Tivoli Identity Manager Server.
- ▶ Workflow properties
Used to configure the core Tivoli Identity Manager workflow engine.
- ▶ Mail properties
Used to configure internal mail notification.

- ▶ Reconciliation properties
Used to configure the reconciliation process where data retrieved from agents is synchronized in the Tivoli Identity Manager database.
- ▶ Shared secret properties
Used to configure the level of protection of the shared secret code.
- ▶ Life cycle rule properties
Define values such as the partition size used for life cycle rules.
- ▶ Product name properties
Identify this product.
- ▶ Application client request properties
Define the properties used to configure the lifetime, or timeout, value for the authentication token used to allow third-party communication with Tivoli Identity Manager Server.
- ▶ Reverse password synchronization properties
Used to configure reverse password synchronization.
- ▶ Post office properties
Used to configure the post office for e-mail collection.
- ▶ Database resource bundle properties
Determine the refresh interval for the database resource bundle.
- ▶ Database cleanup properties
Defines the parameters to clean up session information in the database.
- ▶ Account restore properties
Suppress the need for a new password when an account is restored.
- ▶ Create password check box properties
Define the default check box properties to create a password.
- ▶ Identity feed properties
Define a default identity feed action, such as whether to suspend an account.
- ▶ Upgrade properties
Define values for the upgrade of a given release of Tivoli Identity Manager.
- ▶ Multiple password-synch agent properties
Used to configure the Tivoli Identity Manager Server to support multiple password-synchronization agents.

For more information about system properties located in the `enRole.properties` file, refer to the IBM Tivoli Identity Manager Information Center:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

We encourage you to investigate various configuration options that are written in this file.

Along with the `enRole.properties` file, there are many other configuration files that control various features of Tivoli Identity Manager Server. Some parameters in those files are set up during installation, and we recommend that you do not change those values. We briefly explain some of these configuration files here:

- ▶ `adhocreporting.properties`
The `adhocreporting.properties` supports the custom reporting module.
- ▶ `crystal.properties`
The `crystal.properties` file stores the global properties for the Crystal Reports plug-in of the custom reporting module.
- ▶ `CustomLabels.properties`
The property key and value pairs in the `CustomLabels.properties` file are used by the Tivoli Identity Manager GUI to display the label text for forms. A separate `CustomLabels.properties` file exists for each individual language supported by Tivoli Identity Manager.
- ▶ `DataBaseFunctions.conf`
When designing custom report templates, you can use the database functions with the Report Designer component of Tivoli Identity Manager by defining the functions in the `DataBaseFunctions.conf` file.
- ▶ `enRoleAuditing.properties`
The `enRoleAuditing.properties` file is used to enable or disable the tracking of changes made by a Tivoli Identity Manager user to business objects (person, location, service, and so on) or configuration of the system.
- ▶ `enRoleAuthentication.properties`
The `enRoleAuthentication.properties` file specifies the type of method used by Tivoli Identity Manager to authenticate users and identifies the Java object that provides the specified authentication mechanism.
- ▶ `enRoleDatabase.properties`
The `enRoleDatabase.properties` file specifies attributes that support the relational database used by Tivoli Identity Manager. The data in this file is not intended for manual configuration. Use the `runConfig` utility to set the values for this file.

- ▶ `enRoleLDAPConnection.properties`

The `enRoleLDAPConnections.properties` file provides standard configuration settings that allow successful communication between Tivoli Identity Manager and the LDAP directory server.
- ▶ `enRoleLogging.properties`

The `enRoleLogging.properties` file specifies attributes that govern the operation of the JLog logging and tracing API that is bundled with Tivoli Identity Manager. *JLog* is a logging package for Java that enables you to log messages according to message type and priority, and to control at runtime how these messages are formatted and where they are reported.
- ▶ `enRoleMail.properties`

The `enRoleMail.properties` file contains attributes that specify the mail transport protocol that is used by the JavaMail™ API and other Tivoli Identity Manager application-specific properties. You must provide the values for the application-specific properties. Default values are provided for the JavaMail-specific properties (including the default mail provider and protocol). If you change the default values for the JavaMail-specific properties, you must provide your own testing and verification of your custom protocol and implementation.
- ▶ `enRolepolicies.properties`

The `enRolepolicies.properties` file provides standard and custom settings that support the functionality of the Tivoli Identity Manager provisioning policy. Functionality supported by this properties file includes:

 - Specifying Java classes to process provisioning policy conflicts using join directives
 - Specifying default and non-default join directive caching timeouts
 - Declaring policy attributes to be ignored during policy compliance validation
- ▶ `enRoleWorkflow.properties`

The `enRoleWorkflow.properties` file specifies the XML file mappings for system-defined workflows. The system workflow is identified by a unique type ID and an associated XML file. The XML workflow files are located in the `ITIM_HOME\data\workflow_systemprocess` directory.

You should not remove or modify the default system workflow type IDs and XML file values provided in the `enRoleWorkflow.properties` file.
- ▶ `fesiextensions.properties`

The `fesiextensions.properties` file defines built-in and custom FESI extensions required by Tivoli Identity Manager. FESI refers to the *Free Ecma Script*

Interpreter, a JavaScript interpreter written in Java. The FESI interpreter reads this properties file during Tivoli Identity Manager initialization to set extensions for required Java classes. The FESI extensions represent regions, or hooks, in the Tivoli Identity Manager software where the use of JavaScript code is allowed to introduce built-in or custom business logic.

Note: Because FESI extensions are deprecated, do not author new extensions using this deprecated architecture.

- ▶ `helpmappings.properties`
The `helpmappings.properties` file allows a customer to replace the installed Tivoli Identity Manager help system with an alternative help system.
- ▶ `reportingLabels.properties`
This properties file is similar to other labels-related properties files such as `labels.properties`, or `customLabels.properties`, and holds labels that are used by the Reporting Engine.
- ▶ `reportabledeny.properties`
By default, this property holds a list of Tivoli Identity Manager tables that are used by various Tivoli Identity Manager components to store internal/configuration data for which a report is inappropriate.
- ▶ `scriptframework.properties`
For all new JavaScript extensions, use the `scriptframework.properties` file to configure script extensions and other scripting functions.
- ▶ `SelfServiceHelp.properties`
`SelfServiceHelp.properties` can be used to redirect help to a custom location for customers who want to have their own help content for the self service user interface.
- ▶ `SelfServiceHomePage.properties`
`SelfServiceHomePage.properties` is used to configure the sections of the initially-installed home page for the Self Service UI. You can add or remove tasks, and update icon URLs and labels of the home page from this file.
- ▶ `SelfServiceScreenText.properties`
The `SelfServiceScreenText.properties` file is a resource bundle containing the labels for the self service user interface.

- ▶ SelfServiceUI.properties

The SelfServiceUI.properties file controls miscellaneous properties of the self service user interface.

- ▶ ui.properties

The ui.properties file specifies attributes that affect the operation and display of the Tivoli Identity Manager GUI.

In addition to these files, Tivoli Identity Manager contains more property files that are listed in a table in the IBM Tivoli Identity Manager Information Center:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

Note that not all of the files are configurable and, therefore, must not be modified.

4.15 Modifying system properties with the GUI

You can also modify certain system properties from within the administration console. Unlike previous version, security configuration settings are separated from other general system configuration.

System configuration under the console **Configure System** menu covers the following configuration activities:

- ▶ Manage Service Types
- ▶ Manage Access Types
- ▶ Global Adoption Policies
- ▶ Workflow Notification Properties
- ▶ Post Office
- ▶ Design Forms
- ▶ Manage Entities
- ▶ Manage Operations
- ▶ Manage Life Cycle Rules
- ▶ Configure Policy Join Behaviors
- ▶ Configure Global Policy Enforcement
- ▶ Import Data
- ▶ Export Data

Security settings are located under the **Set Security Settings** console menu and include:

- ▶ Manage Groups
- ▶ Manage Access Control Items
- ▶ Manage Views
- ▶ Set Security Properties
- ▶ Configure Forgotten Password Settings

Further, the menu Set Security Properties covers the following items:

- ▶ Password Settings
 - Enable password editing
 - Hide generated passwords for others
 - Enable password synchronization
 - Set password on user during user creation
 - Password retrieval expiration period in hours
- ▶ Tivoli Identity Manager Login Account Settings
 - Identity account password expiration period in days
 - Maximum number of incorrect login attempts
- ▶ Group Settings
 - Automatically populate Tivoli Identity Manager groups.

4.16 User interface customization

In addition to system configuration, Tivoli Identity Manager provides the possibility to customize the graphical user interface (GUI) and forms representation. Tivoli Identity Manager provides two types of user interface:

- ▶ Administrator console
- ▶ Self-service (user) user interface

Both interfaces are customizable and provide the basic Tivoli Identity Manager functions that users and administrators need.

4.16.1 Administrative console customization

You can customize the administrative console interface in two ways:

- ▶ By using the built-in console framework that takes advantage of ACIs and views.
- ▶ By directly modifying files that are installed within Tivoli Identity Manager, such as properties files and image files, which allow your company to customize the look and feel of the interface.

The key property file for user interface customization is the `ui.properties` file that controls the appearance of the header, footer, and home page and that configures the title, number of pages displayed, and the number of search results returned.

In addition, the `helpmapping.properties` file controls the redirection and mapping of administrative console HTML help.

When customizing the administrative user interface, the following changes are valid and supported:

- ▶ Customizing banner content
- ▶ Customizing footer content
- ▶ Customizing the administrative console home page
- ▶ Customizing the title bar
- ▶ Redirecting help content
- ▶ Customizing the number of items displayed on panels

You can change the home page in the administrative console user interface through customization. The *home page* refers to the main page that is loaded when a user logs in to the administrative console user interface. Section and task definitions connect defined views to tasks and group tasks into sections, which are also called *task panels*.

These section and task definitions are defined in the properties file in the `ITIM_HOME\data` directory. You can code direct links to tasks from the home page to administrative functions.

To update the home page, modify the `ui.homepage.path` key and save the file. Then, enter either a URL of the HTML or JSP file that you use for a home page. You can also put this file in the following directory and prefix the file name with `/itim/console/custom`:

```
WAS_PROFILE_HOME\installedApps\node_name\ITIM.ear\  
itim_console.war\custom
```

If this directory does not exist, you must create it.

The IBM Tivoli Identity Manager Information Center explains the tasks that are supported for direct access:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/cpt/cpt_ic_customize_forms.htm

Forms customization

Tivoli Identity Manager provides default forms to create, view, and modify system entities. System administrators can customize forms for the following system entities using the Tivoli Identity Manager Form Designer:

- ▶ Account
- ▶ Admin domain
- ▶ Business partner organization
- ▶ Business partner persons
- ▶ Tivoli Identity Manager user
- ▶ Location
- ▶ Organization
- ▶ Organizational unit
- ▶ Person
- ▶ Service

The Form Designer enables system administrators to easily manage all entity forms from one location. Only individuals who are part of the administrator group can access this feature.

Each form category folder has associated object profiles that represent system entities. Each object profile is associated with a form template.

Default form templates are generated from an entity's configuration. Form templates have at least one tab and one form element. A tab is a container for grouping form elements. A form element is a system entity attribute. Each tab consists of a label describing the group. Each form element consists of a label describing its data and the data format. Form elements are listed in the order that the elements are presented on the form. The CustomLabels.properties file is used by the Tivoli Identity Manager GUI to display the label text for forms.

Custom Forms in Tivoli Identity Manager can also be used to guarantee the type of data and the syntax of the data users are allowed to enter in fields. This is done through the use of custom constraints. Custom constraints are field-level data restrictions of various types.

For more details about how to configure custom forms, see the IBM Tivoli Identity Manager Information Center:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

4.16.2 Self-service user interface customization

The Tivoli Identity Manager self-service user interface is simplified administration interface for users. It is highly customizable, allowing you to integrate a common corporate appearance while maintaining the flexibility to perform self-care identity management tasks integral to your roles and responsibilities.

You can customize the self-service user interface in two ways:

- ▶ Using the built-in console framework that takes advantage of ACIs and views
- ▶ Directly modifying files installed within Tivoli Identity Manager such as:
 - Properties files
 - Cascading style sheet (CSS) files
 - A subset of Java server pages (JSP) files
 - Image files

You can change the layout in the self-service user interface through customization.

The following files are key files for customization of the self-service interface:

- ▶ `SelfServiceUI.properties`

Controls the layout of the user interface (banner, footer, navigation bar, tool bar), the number of pages displayed, and the number of search results returned. Turning on and off page elements can give a variety of layout options. The only required page element is the content element (content area), which contains the tasks and task panels. To show or hide a page element, change the `ui.layout.showname` property in `SelfServiceUI.properties` file.

The file also allows you to configure the items available in the “Search By” box for user search in the self service interface.

In addition file allows you to enable direct access to the Expired Password change screen and bypass the self service login page under certain conditions. The property key that allows this is `ui.directExpiredChangePasswordEnabled`.

► SelfServiceScreenText.properties

Provides the text that is displayed on the self-service user interface. The following screen text items can be customized:

- Titles
- Subsection titles
- Subsection descriptions
- Field labels
- Table column headers and footers
- Button text

Text that cannot be replaced includes error messages and text in the help content that you access by clicking on the help link. However, it is possible to redirect help requests to a different URL.

► SelfServiceScreenText_language.properties

Provides the language-specific text that displays on the self-service user interface. By default this file is SelfServiceScreenText_en.properties which contains the English language bundle. SelfServiceScreenText.properties is the default file used if no other matching language is found.

► SelfServiceHomePage.properties

Defines the sections of the self-service user interface home page and the order in which they will be displayed.

► SelfServiceHelp.properties

Defines the links to HTML help pages that appear on the self-service user interface. The HTML files are located in the WAS_PROFILE_HOME\installedApps\node_name\ITIM.ear\itim_self_service_help.war directory. You can redirect help by modifying the information in this file.

► SelfServiceScreenTextKeys.properties

Provides label keys that are displayed on the self-service user interface. This file can be used to assist with customization of screen text by providing a template to develop labels and instructions.

The file contains labels which are set to the key name (for example, password_title=password_title).

For more details about how to customize Self-Service interface, see the IBM Tivoli Identity Manager Information Center:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

4.17 Directory server

Tivoli Identity Manager uses an LDAP-based *directory server* to store users, accounts, organization information, and so on. Sometimes, it is necessary to access this information and take some actions *directly* on the directory server without using the Tivoli Identity Manager interface. Therefore, it is very important to know about the detailed organization of the data in the directory server.⁴

The data in the directory server is organized in a tree-like hierarchy known as the *directory information tree* (DIT). Every node has a unique place in the tree, described by its *distinguished name* (DN). Figure 4-4 on page 166 displays the structure of the Tivoli Identity Manager Server DIT, and Table 4-1 on page 166 describes all the nodes in the DIT.

Note: The node `o=OrganizationName` in Figure 4-4 is only a representation that describes this object (node). The real DN for this node is as follows:

```
erglobalida=<some large random number>,ou=<CompanyName>,<Tivoli  
Identity Manager suffix>
```

a. The “erglobalid” is a random number that Tivoli Identity Manager generates upon creation of a new object (for example, organization or user).

For every organization that you create using the Tivoli Identity Manager Web interface within the organization tree, Tivoli Identity Manager generates all branches under that organization, as depicted in Figure 4-4 (for example, `ou=orgchart`, `ou=workflow`, and `ou=policies`).

⁴ It is very common to just use the word *LDAP server* instead of *directory server*. Generally, this usage is a mistake, because LDAP is strictly a protocol that a directory server uses for communication. However, in our day-to-day articulation, the term *LDAP server* or *LDAP database* is definitely an acceptable synonym for *directory server*.

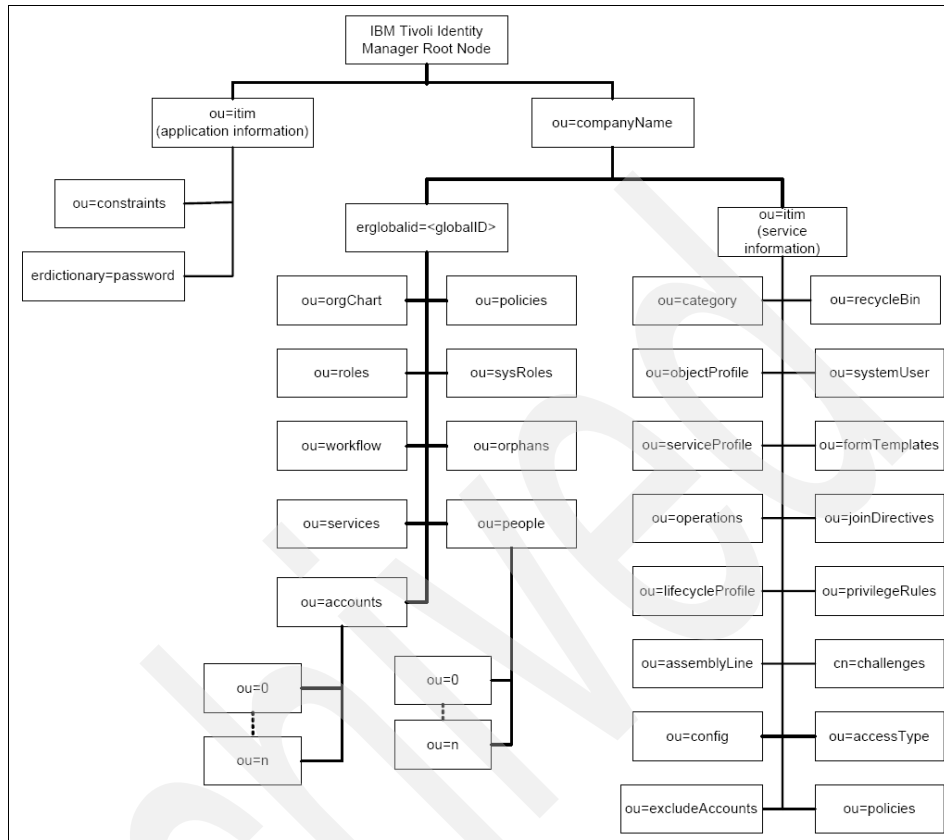


Figure 4-4 Tivoli Identity Manager directory information tree

Table 4-1 Description of components in the Tivoli Identity Manager DIT

Container	Description
Root node	The root node is where the Tivoli Identity Manager Server is installed.
ou=itim (application information)	This container stores all pertinent information for the Tivoli Identity Manager application.
ou=constraints	This container stores membership restrictions for various roles services.
erdictionaryname=password	This container stores invalid password entries for use with password policies.

Container	Description
<i>ou=CompanyName</i>	Name of the company. This container is the parent container for all information pertaining to the company within the Tivoli Identity Manager system.
<i>erglobalid=<GlobalID></i>	This node stores information of the organization. The company long name can be found in this node
<i>ou=orgChart</i>	This container stores the definition of the organizations and organizational units within Tivoli Identity Manager.
<i>ou=roles</i>	This container stores all information for all organizational roles defined within Tivoli Identity Manager.
<i>ou=workflow</i>	This container stores all the workflows designed for the use within the Tivoli Identity Manager system for the company.
<i>ou=services</i>	This container stores information pertaining to the services installed for use with the Tivoli Identity Manager system.
<i>ou=accounts</i>	This container stores all accounts in the Tivoli Identity Manager system.
<i>ou=policies</i>	This container stores all the defined policies.
<i>ou=sysRoles</i>	This container stores all information pertaining to the Tivoli Identity Manager groups defined within Tivoli Identity Manager.
<i>ou=orphans</i>	This container stores all orphan accounts retrieved during a reconciliation.
<i>ou=people</i>	This container stores all information about persons within Tivoli Identity Manager.
<i>ou=itim</i> <i>(service information)</i>	This container is the parent container for system-specific information.
<i>ou=category</i>	This container stores life cycle management operations for an entity type. Only person and account are supported. Global represents the system's operation.
<i>ou=objectProfile</i>	This container stores all information pertaining to the object profile.

Container	Description
ou=serviceProfile	This container stores the service profiles required for the system to recognize a managed resource as a service.
ou=operations	This container stores information about workflow operations (such as add, modify, delete, suspend, and transfer) with Tivoli Identity Manager.
ou=lifecycleProfile	This container stores all information pertaining to the life cycle characteristics that are defined at the entity (instance) level.
ou=assemblyLine	This container stores all information pertaining to the configuration for the service's Tivoli Directory Integrator adapter.
ou=config	This container stores all the information about the workflow configurations.
ou=excludeAccounts	This container stores all the information about which accounts should be excluded during reconciliation.
ou=recycleBin	This container stores entities deleted from the system using the graphical user interface.
ou=systemUser	This container stores information about system users.
ou=formTemplates	This container stores information about the various forms and the form templates used within the system.
ou=joinDirectives	This container stores all the information about the provisioning policy join directives.
ou=privilegeRule	This container stores all information that is used to determine if the difference between an account value and what is dictated by a provisioning policy requires revocation or granting of privileges.
cn=challenges	This container stores all information pertaining to the password challenge/response feature.
ou=accessType	This container stores information about access types.
ou=policies	This container stores information about account defaults for each service.

Along with this tree structure, Tivoli Identity Manager uses directory server (LDAP) classes to describe its entities. Every class is described with one or more attributes. The attributes can be mandatory or optional. Directory server as a standard has many predefined classes and attributes that create an LDAP

schema. For example, in order to describe the entity *person*, Tivoli Identity Manager uses a class named *iNetOrgPerson* and maps all Tivoli Identity Manager person attributes with attributes from the *iNetOrgPerson* class. If some additional attributes (not defined in Tivoli Identity Manager) are required to describe the person, a new custom person entity can be created in Tivoli Identity Manager and mapped to a new class in the directory server. The new class in the directory server can inherit all attributes from the *iNetOrgPerson* class and can be extended with completely new defined attributes that map the new custom person attributes.

In addition, Tivoli Identity Manager has its own classes that are populated into Tivoli Identity Manager during installation. That process is known as *extending a schema design*. For example, one of these extensions is the *erBPPersonItem* class. The *erBPPersonItem* class is an auxiliary class that identifies attributes for a business partner person. It is important to notice that all Tivoli Identity Manager classes begin with the *er* prefix.

To implement the difference between a person and a *BPPerson*, Tivoli Identity Manager uses the standard object class *organizationalPerson* to describe a *BPPerson*. So, if we need to extend *BPPerson* with additional attributes, we need to use the *organizationalPerson* object class.

For more details about the directory server design, refer to the *IBM Tivoli Identity Manager Version 5.0 Database and Schema Reference*, SC32-9011.

Archived



Data management

In this chapter, we describe different approaches for handling the initial load of users into IBM Tivoli Identity Manager. In addition to this chapter, we discuss more issues about every day data management in Chapter 7, “Production” on page 195.

5.1 Identity feed overview

After the initial installation and configuration of Tivoli Identity Manager, one of the most important tasks in a Tivoli Identity Manager deployment is the initial loading of identities into the organization tree. This process is referred to as the *identity feed*.¹

The identity feed uses data from external data sources to create, modify, suspend, and delete person records in Tivoli Identity Manager. Typically, the best-suited authoritative source is the human resources (HR) repository, but in some cases, you need to use multiple sources to pull together the required attributes for a single identity.

There are two types of identity feeds:

- ▶ Initial identity feed, which loads all users into a new *pristine* Tivoli Identity Manager installation.
- ▶ Operational identity feed, which keeps Tivoli Identity Manager users in synchronization with the authoritative source of identities (for example, an HR database).

The identity feed is a critical task and needs to be well planned and designed with several aspects in mind such as the organization tree, the types of users to be loaded, and the method used to load user information from an authoritative source.

5.2 Initial identity feed preparation

After you identify the source for the identity feed, you need to fully understand the organization tree structure and the person attributes that are needed for the initial identity feed. Minimally, Tivoli Identity Manager requires the following information to manage an identity:

- ▶ Common name (directory server attribute CN)
- ▶ Last name (directory server attribute SN)

In addition to these attributes, it is good practice to use as many attributes as possible to describe a person, such as:

- ▶ Title or job description
- ▶ Organizational unit (LDAP OU)
- ▶ Employee number

¹ Identity feed is often referred to as *HR feed*.

- ▶ Department information
- ▶ E-mail address
- ▶ Manager

Note: When we say *collect as many attributes as possible*, we are specifically referring to organizational information. Do not try to create another enterprise whitepages directory that is managed by Tivoli Identity Manager. This is not its goal. Only collect information that is pertinent to the person's position within the organization.

Additional attributes are commonly used to place a user into the appropriate part of the organization tree. Users are placed in the root of the organization tree if no attributes are defined during the initial identity feed that clearly define a location, or attribute values are not recognized (for example, the organization unit is not defined in the organization tree). A Tivoli Identity Manager administrator can later move users to the appropriate part in the organization tree, but this process can be painful if you load hundreds or thousands of users and place them in the root of the tree during the initial identity feed.

As we mentioned previously, identities that are defined in Tivoli Identity Manager are stored in an LDAP directory server. The directory server uses different object classes for *person* and *BPPerson* so that Tivoli Identity Manager can make a distinction between them. Respectively, those object classes are named *iNetOrgPerson* and *organizationalPerson*. Person attributes are mapped to attributes that are defined in those classes. If you need to define additional attributes for a person (that are not defined in the standard class), create a new object class that contains those additional attributes. Then, map this new object class to the *custom person* in Tivoli Identity Manager.

5.3 Types of initial identity feed

After defining user attributes, organization tree, and placement rules, you need to choose one of several approaches of how to load data into Tivoli Identity Manager. The methods and technologies that you use to load the data depend on the following factors:

- ▶ The type of data source.
- ▶ The tools available to extract information from those sources.
- ▶ The policies in place that govern access to the data sources.
- ▶ The programming expertise and other necessary knowledge that can be made available to create a solution for extracting the appropriate data.

You can use several source formats to load identity records into the Tivoli Identity Manager user registry. Tivoli Identity Manager v5.0 has improved the number of common sources for identity feeds that are supported by the product; the following service types are defined to handle the most common sources for identity feeds:

- ▶ Comma-Separated Value (CSV) identity feed
- ▶ DSML identity feed
- ▶ AD OrganizationalPerson identity feed (Microsoft Active Directory)
- ▶ INetOrgPerson (LDAP) identity feed
- ▶ IDI data feed

In addition to using those data sources, we describe a few other methods that you can use to populate your identity store in the following sections.

5.3.1 Manual identity feed

The most simple approach is to define identities manually using the Tivoli Identity Manager Web interface. For large organizations, manually loading person data is not a practical method because thousands of users must be defined. However, you can use the interface to add one or more persons after the initial identity feed is completed.

5.3.2 Comma-Separated Value (CSV) Identity Feed Service

This feed uses a comma-separated value (CSV) file as the source of identity data. A CSV file contains a set of records that are separated by a carriage return/line feed (CR/LF) pair. Each record contains a set of fields separated by a comma. You can use a global identity policy to select the schema attributes that create a user ID.

5.3.3 DSML Identity Feed Service

The Directory Service Markup Language (DSML) Identity Feed Service represents the automated approach. The service can retrieve the information in one of two ways: a *reconciliation* or an *event notification* program. During the reconciliation process, the identities are loaded into Tivoli Identity Manager from a data source defined in the DSML file. An event notification program uses the Java Naming and Directory Interface (JNDI) Service Provider, which is a programming interface to send a notification about changes (events) from a database to the Tivoli Identity Manager Server.

DSML is an XML format that describes directory information. A DSML file represents directory structure information in an XML file format. The DSML file

must contain only valid attributes of the Tivoli Identity Manager profile. The identity feed process uses all objects in the file.

Reconciliation of initial DSML Identity Feed Service

When reconciling the DSML identity feed service, the identity record entries are read from the DSML file and populated into Tivoli Identity Manager. For each identity record entry, the object class is matched to the appropriate person object profile in Tivoli Identity Manager. If a match is made, the distinguished name is converted into a search filter. The search filter looks for an existing match to a person entry that might already exist in the organization that contains the service. If a single match is found, the person data is used to update the existing entry. If no match is found, the individual is added as a new person entry. Duplicate matches return an error and the entry is not added, but the reconciliation process continues running.

If the DSML data feed contains an attribute not mapped to an object class, the user record with that attribute fails in the data feed, but again the reconciliation process continues. To indicate the problems during a reconciliation, a return status of *Warning* in the log. Always check the log files for this situation.

5.3.4 Windows Server Active Directory Identity Feed Service

The Active Directory organizational identity feed provides the capability for creating users based on existing user records in Microsoft Windows Server® Active Directory (AD). This feed allows you to use a directory resource as the source for the feed. Information from the AD organizationalPerson objectclass is mapped to the inetOrgPerson schema. This identity feed loads all user objects under a specified base.

5.3.5 INetOrgPerson Identity Feed

The INetOrgPerson Identity Feed supports LDAP directory server using the inetOrgPerson LDAP objectclass defined in RFC2798. This feed allows you to use a directory resource as the source for the feed. This identity feed loads all inetOrgPerson objects under a specified base. Records that do not have objectclass=inetOrgPerson are ignored.

5.3.6 IDI Data Feed Service

This approach is similar to the DSML identity feed service. In this approach, IBM Tivoli Directory Integrator is used to import identity information into Tivoli Identity Manager. Tivoli Directory Integrator can be programmed to parse information

from separate sources, such as LDAP directories, database tables, Active Directory, Lotus Domino, CSV files, XML files, and more. This provides greater flexibility over the standard data feeds. The IDI Data Feed is provided for instances where the other HR feeds are not sufficient. It provides the ability to define custom identity feeds. Use of this data feed requires knowledge of Tivoli Directory Integrator.

5.3.7 Programming approach

You can also load the identities into Tivoli Identity Manager using the Java Naming and Directory Interface (JNDI) API and an external custom application. This approach requires the knowledge of Java coding.

5.3.8 Self-registration

In some cases, users might be allowed to register themselves using a configurable interface.

5.3.9 Placement rule

Both the IDI Data Feed and DSML Identity Feed Service have a *placement rule* field in the service configuration form. This field can be used to define Java Script in order to position the identities in the organization tree based on attributes imported from the authoritative data source.

5.3.10 Attribute mapping file

The mapping of LDAP attributes to Tivoli Identity Manager attributes is handled by the *attribute mapping file*. The format of the attribute mapping file is

```
feedAttrName=itimAttrName
```

Lines that begin with a number sign (#) or semicolon (;) are interpreted as comments.

The attribute mapping file completely overrides the default mappings. All attributes that are needed from the feed source must be included in the mapping file. The following attributes must be included in the mapping file:

- ▶ Attributes that are specified as required in the person profile form
- ▶ Attributes that are specified as required in the LDAP schema for the target person profile

If an attribute from the feed source is not included in the attribute mapping file, the value is not set on the Tivoli Identity Manager attribute.

5.3.11 Enabling workflow for identity feeds

Regardless of the method that you use for the identity feed, Tivoli Identity Manager can be configured to call the workflow engine to enforce provisioning policies. Enabling the workflow engine results in all applicable provisioning policies being enforced for incoming identities. Enabling the workflow engine results in all applicable provisioning policies being enforced for incoming identities, which can result in slower identity feed performance.

Thus, importing identity records that represent an initial population of people, using the workflow option surely degrades performance. Therefore, have the service provider update the data services directly. Provisioning policies can be applied after the initial population is completed.

5.4 Deploying initial identity feed with existing accounts

When Tivoli Identity Manager is deployed in an environment where accounts already exist on your managed targets, you might want to associate these existing accounts with your newly loaded identities by using an adoption policy.

When creating an adoption rule (part of the adoption policy) you can specify attribute matching using different person attributes and attributes on the target system. Besides, the Java Scripting engine can be used to create more complex rules.

The default global adoption policy assigns an account to a user if the account User ID attribute matches the Tivoli Identity Manager user UID attribute. However, when using Java Scripting and complex adoption rules the other attribute than can help in mapping an account to a person object is the *aliases* attribute defined on the LDAP Person object as *eraliases* multi entry attribute.

An alias is an identity name for a user. A user can have multiple aliases to map to the various user IDs that the user has for accounts.

Accounts that cannot be matched with a user identity are logged as orphan accounts.

After the initial feed has completed, you need to keep Tivoli Identity Manager synchronized with the authoritative source of the identities. Although you can perform this synchronization manually, an automatic synchronization reduces

errors and allows the system to handle several important events that can happen with the identities, such as:

- ▶ New people joining the organization
- ▶ People leaving the organization
- ▶ Changes in positions, jobs, or managers

You need to schedule this procedure according to the business needs, such as daily, hourly, or weekly as business requirements dictate.

This concludes the different approaches for handling the initial load of users into Tivoli Identity Manager.

In the next chapter, we provide information about resources and tools that you can use when identifying and resolving problems related to Tivoli Identity Manager.



Troubleshooting

In this chapter, we provide information about resources and tools that you can use when identifying and resolving issues that might be related to IBM Tivoli Identity Manager.

6.1 Troubleshooting problems

Troubleshooting problems occur due to errors caused by improper installation, configuration, and operation procedures. This section describes basic steps for troubleshooting the various stages of your Tivoli Identity Manager installation and configuration. Later in this chapter, we talk about Tivoli Identity Manager tools that help you find and determine the causes of errors. Those diagnostic tools are:

- ▶ Log files
- ▶ Traces
- ▶ Diagnostic utilities

6.1.1 Troubleshooting installation errors

Installation errors can occur with Tivoli Identity Manager and the middleware software products used. The following sections describe where common installation errors might occur.

Installation errors

Tivoli Identity Manager runs on top of middleware (WebSphere Application Server, supported LDAP, and database servers). To respond to failures that occur when installing and configuring those products, refer to the installation, configuration, and troubleshooting guides that are associated with these products.

Perform the following tasks to respond to errors that occur during the Tivoli Identity Manager installation:

1. Read the message text to determine the source of the problem. Depending on the type of error, the error message might be posted in the installation program window or a command window. If the error is severe, detailed information is saved in a log file. See “IBM Tivoli Identity Manager installation log files” on page 182 for information about the logs that are created during installation.
2. Correct the cause of any errors described in the error message information and retry the installation. Installation errors are also described in *IBM Tivoli Identity Manager Server Installation and Configuration Guide*.
3. Repeat this process until you have concluded that any remaining installation errors are not the result of improper installation or setup.

Database errors

A Tivoli Identity Manager-supported database product must be installed and a database must be created for use by Tivoli Identity Manager before starting the

Tivoli Identity Manager installation program. The DBConfig utility configures the Tivoli Identity Manager database and tables. This utility is started as part of the installation program. It can also be started as a stand-alone program. If the utility is run as a stand-alone program, you must restart the Tivoli Identity Manager Server.

Run this command only if the command failed to configure the database during installation. If the Tivoli Identity Manager database tables have been previously set, running the DBConfig command first drops all previously existing Tivoli Identity Manager tables. Database installation and configuration processing messages are logged in a database log file.

Refer to *IBM Tivoli Identity Manager Server Installation and Configuration Guide* for additional information.

Directory server errors

A supported directory server is used to store current information used by Tivoli Identity Manager to manage identities. A directory server must be installed and operational before the Tivoli Identity Manager schema can be setup. The LDAPConfig utility sets up the schema and default data entries for Tivoli Identity Manager. This utility is started as part of the installation program and can also run as a stand-alone program. If the utility is run as a stand-alone program, you must restart Tivoli Identity Manager.

To avoid the loss of existing directory server data, you must not run this tool manually unless a directory server configuration problem occurs during installation. Directory server installation and configuration processing information is logged in a directory server log file.

Refer to *IBM Tivoli Identity Manager Server Installation and Configuration Guide* for additional information.

WebSphere Application Server errors

Because Tivoli Identity Manager is deployed and configured as a WebSphere enterprise application, deployment and configuration messages are written to the WebSphere administrative console and are logged through the WebSphere Application Server logging facilities.

6.1.2 Troubleshooting operation errors

Information about the various components that process requests and operations is located in the various log files for the Tivoli Identity Manager Server. You can use the information in the logs to determine how a request was handled.

Messages are logged by the Tivoli Identity Manager Server components while handling a task.

In the next section, we discuss the details of the various log files and available tools for diagnosing.

6.2 Log files

Tivoli Identity Manager has logging features that log system events during specific transactions. You can refer to the information contained in the log files to facilitate isolating and debugging system problems. Tivoli Identity Manager uses the IBM Logging Toolkit for Java (JLog) libraries to incorporate message logging and trace facilities.

6.2.1 Types of logs

This section contains lists of the log files that are created after you install and configure the Tivoli Identity Manager Server. Log files contain various levels of information about the processing of the product and other software that is used to complete a task. Log files are grouped into the following categories:

- ▶ Tivoli Identity Manager installation log files
- ▶ Tivoli Identity Manager Server log files
- ▶ Application server log files
- ▶ Web server log files
- ▶ Database log files
- ▶ Directory server log files
- ▶ Tivoli Directory Integrator log files

IBM Tivoli Identity Manager installation log files

You can find installation log files in the following locations:

- ▶ The system temp directory¹
- ▶ The system root directory
- ▶ ITIM_HOME\install_logs directory

The majority of the installation log files are located under the ITIM_HOME\install_logs directory. The temp directory contains the middleware configuration utility log file (cfg_itim_mw.log). The Tivoli Identity Manager Server

¹ Unlike Microsoft Windows, UNIX and Linux systems do not use a temp environment variable for the temp directory. Instead, the /tmp directory is always used as the temp directory on UNIX and Linux systems.

install logs (itim_install.stdout and itim_install.stderr) are located in the system root directory.

IBM WebSphere Application Server log files

Table 6-1 provides a list of the WebSphere Application Server log files that you can use for troubleshooting.

Table 6-1 WebSphere Application Server log files

File name	Directory	Description
SystemOut.log	profile_root/logs/server_name	Contains information about the communication processing between WebSphere Application Server and Tivoli Identity Manager Server.
SystemErr.log	profile_root/logs/server_name	Contains error information about processing between WebSphere Application Server and Tivoli Identity Manager Server.

The following directories are where profile_root is defined on the different operating systems:

- ▶ UNIX
/opt/IBM/WebSphere/AppServer/profiles/profile_name
- ▶ Windows
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile_name

The messages into these files are written by calling Java methods from the following programs that run in WebSphere container:

- ▶ System.out.println()
- ▶ System.err.println().

IBM Tivoli Identity Manager Server log files

Table 6-2 provides a list of the Tivoli Identity Manager Server log files that you can use for troubleshooting.

Table 6-2 Tivoli Identity Manager Server log files

File Name	Directory	Description
access.log	TIVOLI_COMMON_DIRECTOR Y\CTGIM\logs\	Contains information about authentication attempts.
msg.log	TIVOLI_COMMON_DIRECTOR Y\CTGIM\logs\	Contains Tivoli Identity Manager message data.

The Tivoli Identity Manager operational log files are located in subdirectories of the TIVOLI_COMMON_DIRECTORY/CTGIM directory. The default directory for the different operating systems is as follows:

- ▶ Windows systems
C:\Program Files\IBM\tivoli\common\CTGIM\logs
- ▶ UNIX and Linux systems
/opt/IBM/tivoli/common/CTGIM/logs

You can change the default location of the log files during the initial installation and configuration using the installation user interface. After installation and configuration are completed, you can edit the `enroleLogging.properties` file to change the location of the logging files. Tivoli Identity Manager logging properties describes the general logging options and the logging options that are available for each type of logging activity.

For more information about the Tivoli Identity Manager Server log files, see 6.2.2, “Tivoli Identity Manager Server operation log files” on page 186.

Middleware components log files

All middleware components contain the following log files:

- ▶ Web server log files
By default, the DB2 Database log files are stored in the `HTTPServer_install_dir\logs\error.log` directory, for example:
 - Windows systems
C:\Program Files\IBM HTTP Server\logs\error.log
 - UNIX and Linux systems
/opt/IBMIHS/logs/error.log

▶ Database server log files

DB2 Database records database requests in its own log files. You specify the location of these files when you install the database server. By default, the DB2 Database log files are stored in DB_INSTANCE_HOME. for example:

- Windows systems
C:\Program Files\IBM\SQLLIB\DB2
- UNIX and Linux systems
/home/db2inst1

▶ Directory server log files

By default, Tivoli Directory Server installation logs are located in the following directory:

- Windows systems
ITDS_HOME\var
For example C:\Program Files\IBM\LDAP\V6.1\var
- UNIX and Linux systems
ITDS_HOME/var
For example /opt/ibm/ldap/V6.1

By default, Tivoli Directory Server operational logs are located in the following directory:

- Windows systems
ITDS_instance_HOME\logs
For example C:\idsslapd-ldapdb2\logs
- UNIX and Linux systems
ITDS_instance_HOME/logs
For example /home/ldapdb2/idsslapd-ldapdb2/logs

► Tivoli Directory Integrator logs

Tivoli Directory Integrator maintains a log, `ibmdi.log`, that is associated with communications between the Tivoli Identity Manager Server and the agentless adapters (such as Tivoli Identity Manager Server UNIX and Linux Adapter and Tivoli Identity Manager Server LDAP Adapter).

By default, the Tivoli Directory Integrator log files are located in the solution directory for Tivoli Directory Integrator:

- Windows systems

`ITDI_HOME\solDir`

For example `C:\Program Files\IBM\itim\itdi\home\solDir`

- UNIX and Linux systems

The solution directory for the Tivoli Directory Integrator

For example `/opt/IBM/itim/itdi/solDir`

For the ADK-based adapters, the log file is located in the log directory of the adapter. The log file name is `adapternameAgent.log` (for example, the Windows Local adapter is `WinLocalAgent.log`).

For the Tivoli Directory Integrator-based adapters, one log file is used by all the adapters installed on that Tivoli Directory Integrator instance, while for the ADK-based adapters, each adapter has its own log file. It is good to check the log files if you have communication problems between the server and adapter. In a normal situation, you can turn off the adapter logs

6.2.2 Tivoli Identity Manager Server operation log files

When something seems to stop working, the first place to check is the `msg.log` and `traces` (`trace.log` files). They often contain useful information about what is currently being processed and any errors that might have just occurred.

The Tivoli Identity Manager Server log files contain processing activities about Tivoli Identity Manager as an enterprise application. There are two server log files:

- The `msg.log` file contains Tivoli Identity Manager message data.
- The `access.log` file contains information about authentication attempts.

Additional operation file is the security log (`access.log`), which contains information about authentication requests (attempts). The security log is set on by default. You can configure the starting and stopping of the collection of security data, as well as the level of data that is collected and the size of the log file.

The size and number of the message, security, and trace log files are configurable in the `enroleLogging.properties` file. Use the global file size property, `handler.file.maxFileSize`, to change the size of all files. Use the number-of-files property, `handler.file.log_type.maxFiles`, to set the number of files that are maintained for each type of log, before records start to be discarded.

After a set of log files reaches the specified capacity for a single file, the oldest log data is replaced with the newest data. To maintain a longer history of activity, you can specify the number of multiple log files to keep, before data begins to be discarded. The following algorithm is used to manage the number of log files:

1. If the `log_type.log` file reaches 100 KB, the data is moved to another file which is named `log_type1.log`.
2. If the size of `log_type.log` reaches 100 KB again, the data from `log_type1.log` is moved to `log_type2.log` and the data from `log_type.log` is moved to `log_type1.log`.
3. The next time `log_type.log` reaches a size of 100 KB, step 2 is repeated until the maximum number of specified files exist. If the maximum file limit is reached, before the last set of data is moved, the data from `log_typeX.log` is discarded.

The message, security, and trace logs are formatted in XML. Tivoli Identity Manager provides a viewer that enables you to view the log contents as plain text or as formatted HTML.

Message log

The message log contains the Tivoli Identity Manager messages generated during processing. The message log is by default turned on. You can control starting and stopping the collection of message data by changing the following parameter in `enroleLogging.properties` file:

```
logger.msg.logging= [ true | false]
```

Also, you can configure the level of data being collected and the size of the log file.

After the message log file reaches its capacity, starting at the top of the file, data is overwritten—the oldest log data is replaced with the newest data. To maintain a longer history of messages, multiple message log files can be created, and when the `msg.log` file is full, data is moved to another file.

Configuration properties for the message log are located in the `enRoleLogging.properties` file. Changes are not in effect until detected by the Tivoli Identity Manager Server. The Tivoli Identity Manager Server performs periodic checks for updates based on an interval specified in the properties file.

Here are a few message log properties that you can configure in the `enRoleLogging.properties` file:

- ▶ `logger.refreshInterval=300000`

Interval in which the Tivoli Identity Manager Server checks for updates to `enRoleLogging.properties`. This value is specified in milliseconds. The default value is five minutes.

This property is used by the message and server trace logs.

- ▶ `logger.msg.level=INFO`

The message logging level can be:

`INFO` Captures all message types (informational, warning, and error messages).

`WARN` Captures warning and error messages.

`ERROR` Captures only error messages.

- ▶ `handler.file.msg.maxFiles=5`

Specify the maximum number of message log files to be used.

- ▶ `handler.file.maxFileSize=1024`

The maximum size for each log file. Specify the value in kilobytes (KB).

Tivoli Identity Manager messages include the CTGIM prefix, as described in “Message format” in the *Tivoli Identity Manager Messages Guide*.

Access log

The `access.log` file contains information about authentication attempts. The access log file is a text file. You can use any text editor to view the data.

Configuration properties for the access log are located in the `enRoleLogging.properties` file. Changes are not in effect until detected by the Tivoli Identity Manager Server. The Tivoli Identity Manager Server performs periodic checks for updates based on an interval specified in the properties file. Here are a few message log properties that you can configure in the `enRoleLogging.properties` file:

- ▶ `logger.refreshInterval=300000`

Interval in which the Tivoli Identity Manager Server checks for updates to `enRoleLogging.properties`. The value is specified in kilobytes (KB). The default value is 5 minutes.

Note: This property is used by the message and server trace.

- ▶ `logs.logger.msg.com.ibm.itim.security.logChoice=failure`

Type of attempts to be logged:

<code>failure</code>	Only failed attempts are collected.
<code>success</code>	Only successful attempts are collected.
<code>both</code>	Both failed and successful attempts are collected

- ▶ `logger.msg.com.ibm.itim.security.logging=true`

Specify whether or not to collect security events:

<code>true</code>	Turns the security log on.
<code>false</code>	Turns the security log off.

6.3 Traces

Trace data provides more in-depth processing information and is helpful when you are focused on a particular area you suspect is causing a problem. Trace data is more complex and detailed than message data. Depending of the level of tracing, the information can reveal many step-by-step executions of code in the Tivoli Identity Manager application.

There are two traces controlled by Tivoli Identity Manager Server:

- ▶ Server trace
- ▶ Applet trace

6.3.1 Server tracing

The Tivoli Identity Manager trace facility provides methods to capture information about the Tivoli Identity Manager Server internal operations. After the trace log file reaches its capacity, starting at the top of the file, data is overwritten. The oldest log data is replaced with the newest data. To maintain a longer history of messages, multiple trace log files can be created, and when the trace.log fills up, the data is moved into another file.

Configuration of trace log files can be done by changing parameters in the `enRoleLogging.properties` file. As in the case of log files, the following parameters can be configured:

- ▶ Refresh interval
- ▶ Tracing options
- ▶ Log file size
- ▶ Number of log files

For details about configuration parameters, see *IBM Tivoli Identity Manager: Problem Determination Guide*.

The following values for these properties are the default values:

- ▶ `logger.trace.logging=true`
 - `true` Turns on trace logging.
 - `false` Turns off trace logging.
- ▶ `logger.trace.level=DEBUG_MIN`
 - `DEBUG_MIN` The default value and provides the least amount of information.
 - `DEBUG_MID` The minimum level of trace information needed for debugging.
 - `DEBUG_MAX` Provides the most trace information, causing a performance impact. Use this level to narrow down a problem to a specific component.

Note: You can also change the trace logging level using the **runConfig** command.

- ▶ `handler.file.trace.maxFiles=10`

Specifies the maximum number of trace log files to keep before log records start to be discarded.
- ▶ `logger.trace.com.ibm.itim.component_name=tracing_level`

Defines the Tivoli Identity Manager component to be traced.

The *component_name* is the name of the component, and *tracing_level* is the level of tracing to use for that component. The tracing level that is set for a component overrides the inherited level of tracing.

For example, if you set:

```
logger.trace.com.ibm.tim.workflow.level=DEBUG_MAX
```

Then, the `workflow` component is traced at the maximum level of detail (`DEBUG_MAX`) and all other levels continue to be traced at the default minimum level (`DEBUG_MIN`).

Similar to message log, the trace log file is an XML format file. Tivoli Identity Manager comes with viewer tool that produces files formatted in HTML or plain text.

6.3.2 Applet tracing

The applet tracing is handled separately from the Tivoli Identity Manager Server tracing. All applet tracing information is sent to the applet console window on the client.

There are only two entries in the `enRoleLogging.properties` file that control applet tracing:

- ▶ `logger.trace.com.ibm.itim.applet.logging`

Starts and stops applet tracing:

True	Starts tracing.
False	Stops tracing.

- ▶ `logger.trace.com.ibm.itim.applet.level`

Defines the trace logging level:

DEBUG_MIN	The default value and provides the least amount of information.
DEBUG_MID	The minimum level of trace information needed for debugging.
DEBUG_MAX	Provides the most trace information, causing a performance impact. Use this level to narrow down a problem to a specific component.

To view the applet trace data, launch the Java plug-in control panel while the applet is being loaded on the client browser. Then, click **Show Console**. The applet opens, and you see trace data on the console.

6.4 Adapter troubleshooting

Besides server side troubleshooting, very often is necessary to look at the adapter side for the problem solving such as communication between adapter and Tivoli Identity Manager server. Adapter troubleshooting is different for RMI-based adapters (agent less) and ADK-based adapted (agent-based).

RMI-based adapters are developed using Tivoli Directory Integrator AssemblyLines. The major log for logging and traces is `ibmdi.log` that are placed in the solution directory of the dispatcher component.

On the other hand, logs for ADK-based adapters are enabled on the target machines using the `agentCfg` tool. The name of the log file is `adapternameAgent.log`.

Because adapter logs can flood operating system file system, it is wise to turn them off after the problem is resolved.

IBM strategy is to focus on developing new and custom adapters using Tivoli Directory Integrator. So the troubleshooting of custom adapters is focused on troubleshooting AssemblyLines that are loaded in to Tivoli Directory Integrator (note that Tivoli Directory Integrator caches AssemblyLines, so each time you change an AssemblyLine due to testing, it is a good practice to restart your Tivoli Directory Integrator server).

6.5 Diagnostic tools

This section describes useful tools that can assist in troubleshooting and problem determination. These tools are:

- ▶ Audit log
- ▶ Viewer tool for viewing log file data
- ▶ Tivoli Identity Manager serviceability tool

6.5.1 Diagnosing completed requests with the audit log

Use the audit log that the Tivoli Identity Manager user interface provides to diagnose completed requests with adapter communication, policy enforcement, and request approval. Refer to the IBM Tivoli Identity Manager Information Center for more information about setting the audit log option:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

6.5.2 Viewing log file data

Tivoli Identity Manager provides a viewer that enables you to format and view the message log, security log, and trace logs, which are formatted in XML. The viewer produces files that are formatted in HTML or plain text. The viewer can filter message and trace records according to various fields in the records, such

as, time stamp, severity, thread identifier, and component ID. Different types of logs can be combined and viewed together.

To create a single file for presentation, issue the viewer command from a command line window:

- ▶ For Windows systems, use the following command:

```
ITIM_HOME\bin\logviewer\viewer.bat
```

- ▶ For UNIX systems, use the following command:

```
TIM_HOME/bin/logviewer/viewer.sh
```

For the complete syntax and examples, refer to *IBM Tivoli Identity Manager: Problem Determination Guide*, SC32-1494.

6.5.3 Tivoli Identity Manager serviceability tool

The Tivoli Identity Manager serviceability tool assists in gathering information for working with an IBM Software Support representative. The tool collects Tivoli Identity Manager-related log files, performs a check of the product .jar files, gathers some limited configuration details, and creates a compressed file that contains this information. The compressed file can then be transferred or e-mailed to a support representative. Use this tool only when directed to by your support representative.

To start the serviceability tool, use the following command:

- ▶ Windows systems: **ITIM_HOME\bin\win\serviceability.cmd**
- ▶ UNIX systems: **ITIM_HOME/bin/unix/serviceability.sh**

Launch scripts are included for UNIX systems and Windows systems. The launch scripts are configured by the product installation to specify the following three environment variables:

- ▶ WAS_HOME
- ▶ ITIM_HOME
- ▶ PRODUCT_ROOT

Ensure that these variables are set correctly if you encounter problems running the tool.

The following data is captured by the tool:

- ▶ All log files under `.../tivoli/common/CTGIM/logs`
- ▶ MD5 checksum of all .jar files under `WebSphere/AppServer/installedApps/NODE_NAME/enrole.ear/`

- ▶ Java version information
- ▶ Operating system level (including CPU, memory, disk, kernel, and swap information if available)
- ▶ DB2 level (if available)
- ▶ Tivoli Identity Manager information stored in LDAP such as roles, services, workflows, provisioning policies, life cycle operations, service profiles.

6.6 Additional resources

The product documentation covers some of the most common troubleshooting problems and resolutions. The following publications contain problem and solution information:

- ▶ *IBM Tivoli Identity Manager: Problem Determination Guide*, SC32-1561, contains a “Troubleshooting” chapter that describes known problems that can arise during the installation and operation of Tivoli Identity Manager.
- ▶ *IBM Tivoli Identity Manager Version 5.0 Performance Tuning Guide*, SC32-6594, contains a “Troubleshooting” chapter that gives some details about problems with middleware.
- ▶ *IBM Tivoli Identity Manager Server Version 5.0 Installation and Configuration Guide*, SC32-1562, contains a chapter on verifying whether processes are running.
- ▶ *IBM Tivoli Identity Manager Version 5.0: Release Notes*, contains additional problem descriptions and solutions.

Production

Like most enterprise software deployments, the IBM Tivoli Identity Manager deployment usually consists of at least two instances:

- ▶ A non-production (for test and development)
- ▶ A production environment

We recommend that the non-production environment resembles the production environment as closely as possible (including cluster design, agents for all kind of platforms, and so on). This chapter describes the steps necessary to migrate from the non-production environment to the production environment.¹

We also discuss two important Tivoli Identity Manager administrative tasks for the production environment and how to schedule them on a daily, weekly, or monthly basis:

- ▶ Reconciliation
- ▶ Recycle bin maintenance

In a production environment, you also need to run other regular activities, such as maintenance of certificate expiration, backup of data, and auditing and report generation. We do not discuss those tasks in this chapter, because we cover them in other parts of the book.

¹ In addition, IBM recommends that you name the third environment *pre-production*, *QA*, or *staging*. This environment is used to simulate production rollout and to do any performance tuning and testing on the production like data.

7.1 Data migration

In earlier versions of Tivoli Identity Manager, data migration between two systems could only be done manually. This approach was error prone and required additional resources. Tivoli Identity Manager Version 4.6 came with a new feature that automates this migration process. The process of migrating data across Tivoli Identity Manager environments consists of searching for and exporting configured objects from a source server and importing them into a target server. This can be used between development environments, from development to test, from test to production, from production to a disaster recovery site, and so on. In a majority of cases, the most crucial part of migrating or promoting policies and business logic between environments is when moving between the test and production environments.

The Tivoli Identity Manager import/export feature is useful for migrating Tivoli Identity Manager data items and dependant objects between such environments while maintaining data integrity.

The import module is also used to install and load service definition file (service profiles) from a .jar file. However this function is separated from other data import and it is located under **Configure System** → **Manage Service Types** → **Import Service Type**.

7.1.1 Export

There are two types of exports:

- ▶ Partial (selective)
- ▶ Entire (full)

Both types produce a single downloadable .jar file containing an XML file of serialized objects. The .jar file is stored in the Tivoli Identity Manager database until the administrator selects to download the file to the local file system.

The left column of Table 7-1 shows the list of types of data objects that can be exported from Tivoli Identity Manager using this feature.

Table 7-1 Tivoli Identity Manager objects and their exported dependencies

Parent object	Parent object dependencies
Identity Policy Life Cycle Rule Life Cycle Operation	Object Profile
Identity policies Life Cycle Rule Life Cycle Operation Password Policy Provisioning Policy Service Service Selection Policy Workflow	Service profile
Provisioning Policy Workflow	Organizational Role
Adoption Policy Identity Policy Password Policy Provisioning Policy	Service
Life cycle rules	Life cycle operation

In order to guarantee the integrity of the data throughout the migration process, Tivoli Identity Manager automatically detects and includes any dependencies an exported object might have. A dependency is generally an individual object referenced by a parent or root object that is required on a target system to successfully import the parent.

Dependencies can be classified as functional or logical. A *logical dependency* is a dependency enforced by the schema and will be exported with a parent object. A *functional dependency* is an object required for the Tivoli Identity Manager functionality and will not be exported during a partial export.

Performing a partial export of individual items might not actually export all of the dependencies needed for the object to function. Export only saves the dependencies needed to create the object being saved. It does not ensure that it will execute.

For example, the *service instance* parent object includes the *service profile* and the *service definition*. Therefore, the service profile is a logical dependency. The

provisioning policy object, when exported, does not include functional objects such as an *identity policy* or a *password policy*.

The right column of Table 7-1 shows dependencies for every type of data object that can be exported from Tivoli Identity Manager using this feature.

Exporting everything through the use of the full export will save all of the data supported by the export in the system.

7.1.2 Import

The import process is initialized by an administrator on a target server after extracting objects (generating an export .jar file) from a source server. The import process consists of several individual stages that enable the administrator to interact with and configure the process:

- ▶ .jar file upload
- ▶ Difference evaluation
- ▶ Conflict resolution
- ▶ Data commit to the system

After the .jar file is uploaded, the import process evaluates differences between the data imported and the data in the target server and helps resolve conflicts between the two. Difference evaluation generates a list of objects that are found in the import .jar file and in the target system so that administrators can resolve conflicts on a per object basis by deciding precedence over existing data or by overwriting existing data with the import data. The data is then committed to the system after the conflicts are resolved by the administrator.

Policy enforcement

Importing provisioning policies and dynamic organizational roles might result in associating different people with new roles. Imported policies that have changes that require re-evaluation might result in the following policy enforcement tasks:

- ▶ Evaluating dynamic role changes and updating role memberships
- ▶ Finding provisioning policies associated with host selection policies
- ▶ Combining these with policies that are being imported
- ▶ Enforcing policies on all affected users through a new workflow process

7.1.3 Additional considerations

There are a few key assumptions implicit in the data structure of exports and in the import process logic that are important to the successful completion of an import.

Keep in mind that functional dependencies are not calculated during selective export (only logical dependencies). If you are selectively exporting objects and you have functional dependencies on other objects that are not required to “define” the object, you will need to include it in the selection formula. For example, the default *identity policy* is not be exported with a *provisioning policy*, or a *service*.

After ensuring all relevant service profiles have been installed prior to the import, the target server must be further prepared by creating an organizational structure identical to the one found on the source server. Because objects in the organizational tree are not included in the export data (locations, organizational units, and administrative domains are all excluded), container names are key to a successful import. Container names are saved as references in parent and dependent objects, which the import process uses to look for containers in order to reestablish the object hierarchy in the target server. The profiles, life cycle rules, and life cycle operations are attached to the organization node. Because of this, the name of the organization needs to be matched with the default organization short name to successfully import these objects. In addition, dependent objects (such as service owners and workflow participants) are omitted from the export data and those objects must be found in the target server in order for the import process to successfully reestablish the link for those objects.

Refer to “Developing workflows and policies” in the IBM Tivoli Identity Manager Information Center for more details:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

7.2 Reconciliation

Tivoli Identity Manager is a dynamic system, because users and resources are constantly changing within a company. New people are joining the company, some are leaving, and many are moving inside the company to different positions. With this dynamic user life cycle comes the frequent need of changing access rights to resources. Also, business procedures and policies can change, and there is the need to constantly check those with Tivoli Identity Manager policies.

The reconciliation function synchronizes Tivoli Identity Manager user information with the distributed user accounts on managed resources. During reconciliation, any account created outside of Tivoli Identity Manager (for example, an account created manually on a Linux resource by a local administrator) is discovered and inserted into the Tivoli Identity Manager directory. Tivoli Identity Manager then

tries to match the accounts discovered during the reconciliation process with Tivoli Identity Manager users by using adoption rules. An account that cannot be adopted is labeled an orphan account.

Along with the adoption process, the reconciliation function enables you to detect any policy rule violation and, therefore, identify areas in your organization that are not compliant with security policies. The reconciliation can be executed manually, but the best way is to reconcile your data on a scheduled basis. Let us take a closer look at the scheduling options.

Consider the following best practices for using reconciliation:

- ▶ Perform supporting data reconciliation separately from accounts. The separation is useful during initial deployment for the service and also useful for sync up changes of metadata without accounts, which is very time-consuming. Supporting data includes group configuration information, which contains key information about access privileges on the resource. Bringing back the group data ahead of time allows policies to be configured promptly before accounts are reconciled, so that the policies can be enforced.
- ▶ Set up reconciliation schedules appropriately based on the frequency of data changes. Leave enough time between two reconciliations. Avoid unnecessary reconciliations.
- ▶ Queries are used to break reconciliation into smaller packets. Reconcile only the data that is changed by using Query. Reconciliation is an expensive process, especially when policy checking is enabled.
- ▶ If you are working with a large data repository (that is, a large number of accounts), consider using Query to segment the data and perform the reconciliation in smaller chunks on different schedules.
- ▶ Specify a subset of account attributes to bring back to improve performance.

Scheduling reconciliation

Reconciliations can be scheduled at specific times and configured to return or exclude specific parameters. The configuration of scheduled reconciliation has several pages with different options. Let us walk through the scheduled reconciliation parameters.

The configuration has four major pages:

► **Manage Schedules** page

On the Manage Schedules page, you can configure the following parameters:

- Specify whether a policy evaluates the accounts that the reconciliation returns. Enabling this option can affect performance, but it adds value if you want to enforce policy compliance.
- Use the Create/Change/Delete buttons to manage new and existing reconciliation schedules.

After you select to create or change a reconciliation schedule, you can define more reconciliation parameters using the following pages:

► **General** page

- Display name: The title of created reconciliation schedule.
- Description: Optional field to provide more detailed information about your reconciliation schedule.
- Lock service during reconciliation (yes/no).

The *Lock Service During Reconciliation* check box enables system administrators to lock a service when a reconciliation is performed.

If the Lock Service During Reconciliation check box is selected, any requests sent to the service during a reconciliation are marked as *pending* and are saved in the remote services request queue and processed after the reconciliation is complete.

If the Lock Service During Reconciliation check box is not set, requests issued to the service during a reconciliation will be immediately sent to the service. The Tivoli Identity Manager LDAP directory will also be updated by the requests. To ensure that subsequent reconciliation results do not overwrite these directory updates, a temporary list of these requests is also stored in the local database. Each account returned by the reconciliation is then checked against this temporary list of updates to see whether it is for an account that has been modified during the reconciliation. If so, the reconciliation data for the account will be amended to reflect the modification before being processed further.

This checking can result in a considerable performance impact, particularly for services with a large number of accounts. Therefore, enabling Lock Service During Reconciliation option helps in improving performance.

- Maximum duration (minutes).

The Max Duration option specifies the maximum amount of time a reconciliation is allowed to run before the system forces the reconciliation to end. This value is in minutes and by default it is set to 600 minutes.

► **Schedule** page

On this page you can specify the date and time for the reconciliation.

► **Query** page

This page is an optional page and you can specify if you are performing a “supporting data only” reconciliation, which only returns metadata for accounts and excludes accounts, or use the LDAP filter to specify the subset of accounts or specific type of support data such as a group to be included in the reconciliation.

Specify the subset of account attributes to return during the reconciliation. By default, Tivoli Identity Manager brings back all attributes of accounts. By specifying a subset of attributes that are likely to be changed on the remote resource, you can improve reconciliation performance. In other words, the LDAP filter can define what is or is not included in the reconciliation.

7.2.1 Reconciliation of manual service

The service instance creation steps allow you to perform a reconciliation of a manual service using a comma-separated value (CSV) file that you provide. The reconciliation populates Tivoli Identity Manager with accounts and groups that exist on the manual service. The CSV file contains group and account information. You can provide the reconciliation file at service creation time or at any time the service is modified. There is also a supporting data only option for reconciliation that is used when you want to pull group information from the CSV file, but you do not want to touch accounts in Tivoli Identity Manager.

7.3 Recycle bin periodical maintenance

One of the nodes in the directory server is *ou=recycleBin*. This object can only be accessed from an LDAP browser (there is no appropriate form in the Tivoli Identity Manager Web interface) and usually needs to be periodically maintained.

The recycle bin is disabled by default, but can be enabled by editing the `enRole.properties` file in the `ITIM_HOME\data` directory.

```
enrole.recyclebin.enable = [true | false]
```

If recycle bin is enabled, all deleted objects in Tivoli Identity Manager (such as organization units, persons, accounts, and so on) that are not permanently

removed from the system get moved to the Tivoli Identity Manager *recycle bin* container under the following DN (see Figure 4-4 on page 166):

```
ou=recycleBin,ou=itim.ou=<company name>, <Tivoli Identity Manager  
suffix>
```

The regular cleanup of the objects from the recycle bin is a separate deletion process that involves running cleanup scripts.

During installation (or later by running the runConfig utility), you can change the *Recycle Bin Age Limit* parameter that specifies the number of days that an object remains in the recycle bin of the system before it becomes available for deletion by cleanup scripts. The cleanup scripts can only remove those objects that are older than the age limit setting. For example, if the age limit setting is 62 days (the default value), only objects that have been in the recycle bin for more than 62 days can be deleted by cleanup scripts.

You can use the following scripts to either manually remove or to schedule the periodic cleanup of recycle bin entries with expired age limits:

► Windows platform:

```
ITIM_HOME\bin\win\ldapClean.cmd
```

To schedule periodic cleanup, register the prior command script with the Windows scheduler.

► UNIX platform:

```
ITIM_HOME/bin/unix/ldapClean.sh.
```

To schedule periodic cleanup, create a UNIX cron job. For an example, see:

```
ITIM_HOME/bin/unix/schedule_garbage.cron
```

In addition, there might be business requirements to archive the objects in the recycle bin before they are deleted. They can be archived into another LDAP directory or into an LDIF file.

Archived



Maintenance

Every IBM Tivoli Identity Manager system is somehow unique, and there are a large number of parameters that can have an impact on its functionality. In this chapter, we provide an overview of performance monitoring and tuning of Tivoli Identity Manager. We also discuss the necessary steps for fix pack installation and for upgrade procedure.

8.1 Performance monitoring and tuning

The Tivoli Identity Manager environment consists of many components and parameters that can have an impact on the overall performance of the system. There is no one single approach to this solution, because the parameters depend on many factors that are specific to every Tivoli Identity Manager installation, such as:

- ▶ Tivoli Identity Manager installation size (number of users, workflows, and so on)
- ▶ Single or clustered IBM WebSphere Application Server environment
- ▶ Type of directory server and database that store all the information
- ▶ Type of operating systems
- ▶ Networking infrastructure
- ▶ Hardware utilization (CPU, hard disk, memory, and so on)

Because every Tivoli Identity Manager installation situation is unique, in order to optimize the system, we can demonstrate some best practice steps common for all installations. It is necessary to monitor various system parameters that can have an impact on the system performance.

For all your performance related work, you should generally consult the *IBM Tivoli Identity Manager Version 5.0 Performance Tuning Guide*, SC32-6594. Another great source for Tivoli Identity Manager performance (and other) related topics is the Tivoli Identity Manager Wiki, which can be accessed at:

<http://www.ibm.com/developerworks/wikis/display/tivoliim/Home>

8.1.1 Performance monitoring

Start by monitoring the processor and disk usage of every server, including WebSphere Application Server (Tivoli Identity Manager application), directory server, and database server, to see which server is most heavily used. Based on this information, review the monitoring and tuning steps specific to that component.

The Tivoli Identity Manager application makes intense usage of both the database and the directory server. The database is used as an information staging area and audit trail for provisioning actions, and the directory server is used as a permanent storage location and can be heavily queried when evaluating provisioning policies. Because of this usage, it is possible for either the database or the directory server to be a bottleneck during heavy usage or large provisioning changes. Usually, the first step in identifying a bottleneck in performance is to start monitoring the processor and disk usage of every server to see which server is most heavily utilized.

There are two kinds of performance monitoring for a Tivoli Identity Manager system:

- ▶ Initial performance monitoring after a successful installation and configuration of the product. This monitoring can be regarded part of a final health check on the functionality of all the components of a Tivoli Identity Manager installation. Depending on your monitoring results and the problematic areas you observed (for example, slow HR data feed), various tuning steps can be performed.
- ▶ Periodical monitoring of the system that includes various tasks such as:
 - Reviewing the log files of Tivoli Identity Manager Server and middleware components.
 - Checking utilization and functionality of hardware components (CPU utilization, storage space, and so on).
 - Maintenance of network connectivity between Tivoli Identity Manager Server components, such as adapter connectivity.
 - Monitoring database data and update database statistics using **runstats**.
 - Monitoring directory server data and recycle bin maintenance, as described in 7.3, “Recycle bin periodical maintenance” on page 202.
 - Backup procedures,
 - Evaluate and applying fixes, and so on.

8.1.2 Tuning

Tuning is the process of optimizing the Tivoli Identity Manager system by changing various numbers of parameters. The following recommendations can help guide you in setting up your environment.

Hardware

Consider the following hardware recommendations:

- ▶ Database and directory activity can be very *CPU- and memory-intensive*. It is recommended that each application have at least one processor and 2 GB of RAM each. More processors are generally better. To obtain optimal performance, it is not recommended to have all Tivoli Identity Manager components on the same system unless it is a very powerful unit such as an 8 x 2.6 GHz server with 12 GB RAM.
- ▶ In general, *network latency* is not a major performance bottleneck, but it is still a good idea to have as few hops as possible between components. This includes the Tivoli Identity Manager Server components, the directory server, database server, agents, and agent endpoints. Ideally all components should be on the same subnet or no more than one hop away and on a 100 Mb or faster network.

- ▶ When working with *LPARs*, the following items can improve performance (the list items that improve performance the most are listed first):
 - SMT should be disabled for Tivoli Identity Manager nodes.
 - Have the latest update of the WebSphere Application Server JVM (SR6 improves performance on LPARs).
 - Give at least 1 physical CPU to each LPAR.
 - Use dedicated instead of virtual processors.
- ▶ To minimize *hard disk* bottlenecks, it is frequently better to have several disks in the system rather than one large drive. IBM DB2 has the ability to use multiple disks but must be configured to do so. Keep in mind that high-end I/O backplanes or other advanced storage systems can overcome this by balancing the load across multiple disks automatically.
- ▶ When putting data on a *SAN* in a failover environment, ensure that separate physical devices in the SAN are used for each failover's underlying datastore. If, for example, the database for each LDAP server is on the same physical devices you will likely introduce I/O performance problems.

Software

Consider the following software recommendations:

- ▶ Each agent modifies the LDAP schema by adding new attributes to support a new service. These attributes are created without *indexes*, and for services that service thousand of users, a large benefit can be achieved by adding indexes to attributes with many members.
- ▶ Complicated *provisioning policies* can result in complicated directory and database queries with poor performance. Policies with small numbers of roles and services will perform best.
- ▶ Provisioning policies without *account approval workflows* perform better than those with account approval workflows due to optimizations for the former case.
- ▶ Provisioning policies created by the system when a service is created use a *Default Account Request Workflow*. This account workflow should be removed from the provisioning policy if it is not needed to improve provisioning policy performance.
- ▶ *Dynamic roles* affect people in a given scope, either one-level or subtree. When a person object within that scope is modified or added, that role must be re-evaluated. This is true for every dynamic role in the system. For instance, if there are three dynamic roles with subtree scope and a person object within that scope is updated, all three dynamic roles will be re-evaluated. For this reason, it is recommended that you limit the number of dynamic roles, either by number or by scope, that affect people that are modified frequently. It does not matter if the dynamic role ends up enrolling

the person or not, the evaluation itself is the performance-impacting overhead.

- ▶ When creating *dynamic roles* that apply to all people within an organizational unit, place the dynamic role inside the organizational unit and use the filter (`objectclass=*`). This will yield better performance from the directory server than a filter such as (`cn=*`).
- ▶ Limiting the scope (through placement within the organizational tree) and number of *ACIs* will increase performance by requiring fewer evaluations. When doing a person search through the APIs, be sure to limit the scope of your search to be as narrow as possible to avoid the system evaluating more ACIs than necessary.
- ▶ Only store *person information* needed for policy evaluation and account management within Tivoli Identity Manager to reduce attribute updates that are not used for policy enforcement. When a person object is updated either manually or using some automated method such as an HR feed or JNDI update, all provisioning policies that the user is a member of must be re-evaluated to see if the update would change a provisioning action. For this reason it is recommended to minimize person updates when possible.
- ▶ When loading users into Tivoli Identity Manager (*HR feed*), particularly in bulk when using DSML file or through an IDI data feed, do not include O, OU, or L attributes on the person records. Tivoli Identity Manager includes searches on these attributes for the organizational chart, which can slow down if large numbers of users have these attributes as well. This is particularly important for larger user populations.
- ▶ When loading objects such as people or accounts directly into the Tivoli Identity Manager LDAP server, such as during an initial data load using LDIF, use all numeric values for the `erGlobalID` rather than an alphanumeric value. Numeric *erGlobalIDs* allows the application to make more efficient use of memory when processing reconciliations.
- ▶ When loading a test environment, ensure users have unique IDs for their surname (`sn`) attribute as this is used by the *default identity policy* to determine a unique UID for an account. Having the same value for all users will result in very poor performance when the identity policy creates a new ID for a service due to the resulting LDAP lookups.
- ▶ As a recommendation, each person should have no more than 1200 accounts. Often *administrative accounts* on target systems are mapped to a single person object resulting in one person with possibly thousands of accounts, which can degrade performance or result in Java OutOfMemory errors. The ability to scale beyond this number will depend on the specifics of the system configuration and hardware.

- ▶ Complex *workflows* (either operational workflows or account/access request workflows) can degrade performance. Keep frequently-used workflows, such as the modify person operational workflow, as simple as possible.
- ▶ In a workflow, each transition results in a message being placed on the JMS queue. Design workflows such that they have the fewest number of transitions from start to finish as possible. Consider reducing the number of nodes by combining adjacent scripts nodes. In some cases it might be desirable to create an initial node at the beginning of a long workflow to jump to a specific node later thereby bypassing several unnecessary transitions.
- ▶ If the workflow does several poorly-performing, non-cached, LDAP lookups, it might perform better if the result of the redundant lookups were stored as relevant data and reused in later nodes.
- ▶ Keep the quantity and size of workflow relevant data objects as small as possible. Relevant data must be serialized and deserialized from the database for each node transition.

There are large numbers of different parameters written in various configuration files of the Tivoli Identity Manager Server. We have described some of those parameters here, but for more details, refer to the Tivoli Identity Manager documentation, especially the IBM Tivoli Identity Manager Information Center:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

For more details about some specific tuning tasks, refer to *IBM Tivoli Identity Manager Version 5.0 Performance Tuning Guide*, SC32-6594.

8.2 Migration

One of the maintenance tasks is to keep the system up to date with the latest fix packs and updates. The Tivoli Identity Manager installation program supports upgrades to Tivoli Identity Manager V5.0, from

- ▶ Tivoli Identity Manager V4.5.1 with minimum Fix Pack 36.
- ▶ Tivoli Identity Manager V4.6 with minimum interim fix 47.

The upgrade process is not completely automatic and needs to be well planned. Manual steps might be required to preserve data before an upgrade, as well as a re-customization of settings afterwards depending on each deployment. In addition, the operating system and Tivoli Identity Manager prerequisite middleware components must be manually upgraded to the supported versions. We describe the minimum software-level requirements in “Supported software and operating systems” on page 82, but it is always a good practice to consult

the *IBM Tivoli Identity Manager Version 5.0: Release Notes*, before making any changes.

8.2.1 Migration planning and preparation phase

The migration of a Tivoli Identity Manager environment should not be executed before good planning and preparation. The Tivoli Identity Manager documentation should be reviewed to understand the upgrade process. Operating system and middleware components need to be upgraded manually before the Tivoli Identity Manager Server upgrade process starts. Also, the upgrade process of the Tivoli Identity Manager Server is not completely automatic, and there are some post-migration tasks that need to be performed. The Tivoli Identity Manager installation program upgrades the following components:

- ▶ Database schema and data
- ▶ Directory server schema and data
- ▶ WebSphere Application Server configuration for Tivoli Identity Manager
- ▶ Tivoli Identity Manager configuration files
- ▶ Tivoli Identity Manager application binaries

Notice also that as a preparation phase, all activities need to be minimized and a backup of all system data is recommended.

To upgrade the middleware components, consult the product documentation for every component. Notice that the WebSphere installation and upgrade is not included in the Tivoli Identity Manager Server installation and that upgrade needs to be performed in advance.

8.2.2 Tivoli Identity Manager Server upgrade phase

The upgrade process consists of the following major tasks.

1. Migrate your operating system to a level that this release of Tivoli Identity Manager supports, and ensure that the system has the required fix pack or patches installed.
2. Migrate your database to a supported Tivoli Identity Manager Version 5.0 database, and ensure you can execute database commands.
3. Migrate your directory server to a supported Tivoli Identity Manager Version 5.0 directory server, and ensure you can execute directory server commands.
4. If you are using IBM Tivoli Directory Integrator, migrate to a supported Tivoli Identity Manager Version 5.0 directory integrator.

You might want to uninstall all Tivoli Identity Manager adapters from Tivoli Directory Integrator prior to the upgrade.

5. Install WebSphere Application Server Version 6.1 in a separate directory. Running WebSphere Application Server Version 6.1 upgrade utilities (WASPreUpgrade and WASPostUpgrade) is not recommended.
6. Stop the old version of WebSphere Application Server where Tivoli Identity Manager is running.
7. Upgrade the Tivoli Identity Manager Server using the Tivoli Identity Manager Version 5.0 installation program. The Tivoli Identity Manager installation program upgrades the database schema and data, the directory server schema and data, the WebSphere Application Server configuration for Tivoli Identity Manager, the Tivoli Identity Manager property files, and other Tivoli Identity Manager files. During the upgrade process, the ITIM_HOME\data directory is backed up to the ITIM_HOME\data\backup directory for later recovery if necessary.

Thus, if you are planning to perform the upgrade procedure on a separate new system, you should first copy the Tivoli Identity Manager Version 4.5.1 or 4.6 home directory to the target operating system environment.

After you upgraded successfully, you can validate the current Tivoli Identity Manager version by examining the copyright notice in the header of the Messages.properties file in the ITIM_HOME\data directory.

If you are upgrading Tivoli Identity Manager in a cluster environment, you first need to upgrade Tivoli Identity Manager on the WebSphere Application Server Deployment Manager to upgrade the database and directory server before upgrading Tivoli Identity Manager on cluster members.

8.2.3 Post-upgrade phase

At the end of the migration process, you need to perform the following manual customization tasks, because the Tivoli Identity Manager Server does not preserve all customization of the previous version:

- ▶ The upgrade process does not preserve the following workflow processes, which you must stop or allow to complete before you upgrade Tivoli Identity Manager:
 - Policy Add/Modify/Remove
 - Dynamic Role Add/Modify/Remove
 - Reconciliations
 - Identity feeds

- ▶ Java security
You need to manually apply the changes that you made for the previous IBM Development Kit for Java to the new IBM Development Kit for Java bundled with WebSphere Application Server 6.1.
- ▶ Custom logos used in a Welcome page and XLS style sheets.
If you modified the welcome page, you must re-implement the Styles.css file.
- ▶ EJB™ user ID and password (for Tivoli Identity Manager 4.6 on WebSphere Application Server 5.1)
During the upgrade you have to enter the WebSphere Application Server 6.1 administrator user ID and password.
- ▶ Any customized WebSphere Application Server configurations.
- ▶ Crystal configuration
You should back up all existing Crystal configuration scripts before performing the upgrade so the same scripts can be referenced later.

Additionally, you must manually upgrade the following components.

- ▶ Tivoli Identity Manager jar files that the Tivoli Identity Manager client applications use.
Tivoli Identity Manager client applications must replace their Tivoli Identity Manager Version 4.6 `itim_api.jar`, `api_ejb.jar`, `itim_server_api.jar` and `jlog.jar` files with those from Tivoli Identity Manager Version 5.0.
- ▶ Recertification policies¹
Tivoli Identity Manager recertification policies replace their Tivoli Identity Manager Express Version 4.6 equivalents.
- ▶ Notification templates
The upgrade program does not overwrite (upgrade) any notification templates if you have updated the default notification templates from Tivoli Identity Manager Version 4.6. To migrate old notification templates to Tivoli Identity Manager Version 5.0, you must update both the XML Text Template Language (XTTL) contents and the style.

¹ This is for Tivoli Identity Manager Express only because the Tivoli Identity Manager V4.x does not use recertification policies.

► JavaScript extensions

Tivoli Identity Manager supports two different Java Script interpreters:

- IBM JSEngine
- Free EcmaScript Interpreter (FESI, now deprecated).

Both of these interpreters support the 3rd edition (December 1999) of the ECMA-262 specification and are heavily used to customize workflows. FESI was the first choice for Tivoli Identity Manager V4.6, but it has been deprecated for Tivoli Identity Manager V5.0. Old FESI extensions written for Tivoli Identity Manager V4.6 will still work, but IBM does not recommend writing new extensions using this architecture. After migration to V5.0 consider to migrate all FESI extensions to JSEngine, because the written code will be shorter, and easier to read and understand. If you still need to keep old written FESI extensions in the new environment, keep in mind that configurations are kept in different files: `fesixtensions.properties` versus `scriptframework.properties`.

The `scriptframework.properties` file comes pre-configured to load the extensions necessary to use Tivoli Identity Manager with its default scripts. Do not remove any lines in `scriptframework.properties` because removal could cause Tivoli Identity Manager to stop functioning properly.

The most heavily used section of the property file is for configuring which extensions to load for each host component. To have the script framework load an extension, add a key-value line to the `scriptframework.properties` file that is similar to this example:

```
ITIM.extension.{Host Component}=com.ibm.itim.class_name
```

Adapters

The adapter version follows the version of the Tivoli Identity Manager Server. This means that for Tivoli Identity Manager V5.0 adapters have to be on that level, too. The exception is if you migrated Tivoli Identity Manager to V5.0. Old adapters (using old 4.5.1 and 4.6 profiles) will function in Tivoli Identity Manager V 5.0, but they are supported only during an upgrade. You can even import 4.6 profiles to Tivoli Identity Manager V5.0, but it is not supported.

The recommendation is to perform an upgrade of all adapters after completing the upgrade process of the Tivoli Identity Manager Server. The opposite way will not work because Tivoli Identity Manager 4.x profiles cannot communicate with Tivoli Identity Manager 5.0 adapters.

ADK-based adapters upgrade

Due to a new encryption algorithm in ADK 5.0, you must uninstall the 4.6 version of the adapter before you can install the 5.0 adapter. ADK adapter upgrades consist of two steps, upgrading the adapter software that runs on a target system

and of the adapter profile (service definition files) that sits on your Tivoli Identity Manager Server using the Import function.

IBM Tivoli Directory Integrator-based adapters upgrade

Tivoli Directory Integrator version 6.1.1 is a prerequisite to run the Tivoli Identity Manager 5.0 adapters. To upgrade the adapter from a Tivoli Identity Manager 4.6 installation, perform the following steps:

1. If the Tivoli Directory Integrator is not version 6.1.1, take the appropriate actions to upgrade it.
2. Install all the adapter's components, as described in the installation guide, on the Tivoli Directory Integrator version 6.1.1. The installer will replace any previous installation.
3. Import the adapter profile (service definition files) into the Tivoli Identity Manager 5.0.

Do not forget to restart Tivoli Directory Integrator as the new adapter profile (together with AssemblyLines) is located in directory server and Tivoli Directory Integrator keeps a local cache.

8.3 Fix pack installation

A part of regular maintenance is to keep your system up to date with current levels of application fixes. As a WebSphere Application Server application, Tivoli Identity Manager Server 5.0 uses the WebSphere Application Server fix pack installation wizard called WebSphere Update Installer. This wizard requires that Tivoli Identity Manager application fix packs and interim fixes are packaged in the .pak format. To perform a fix pack installation requires the following prerequisites:

- ▶ WebSphere Update Installer has to be available on the system.
- ▶ For a single server environment
 - IBM WebSphere Application Server must be stopped
 - LDAP and DB2 must be started
- ▶ For a server on a WebSphere Application Server cluster
 - Node agents must be started
 - The Tivoli Identity Manager and Tivoli Identity Manager JMS clusters must be stopped
 - LDAP and DB2 must be started

As a good practice, you always want to perform a backup of critical data (LDAP, DB2, WebSphere and Tivoli Identity Manager Server and adapter configuration files) before applying a fix pack. Test the fix pack in a non-production

environment to avoid unexpected behavior with your customized Tivoli Identity Manager environment.

In addition to the Tivoli Identity Manager Server, it is recommended to keep other components of the system up to date as well, such as WebSphere Application Server, directory server, database, Tivoli Directory Integrator, and adapters.



Sample questions

This appendix provides sample questions for Test 934.

Questions

The following questions can assist in studying for Certification Test 934:

1. Which option would be most appropriate to include in a life cycle management design?
 - a. Provisioning policy definition
 - b. The requirements for dynamic role definition
 - c. Reconciliation requirements for Active Directory
 - d. The requirements for how often to check for inactive accounts
2. Which relationship must be understood in order to create a dynamic role?
 - a. The relationship between dynamic role and static role
 - b. The relationship between dynamic role and person attributes
 - c. The relationship between dynamic role and provisioning policy membership
 - d. The relationship between dynamic role and IBM Tivoli Identity Manager groups
3. Which two of these join directives can be used when multiple provisioning policies affect the same account? (Choose two.)
 - a. Xor
 - b. Not
 - c. And
 - d. None
 - e. Union
4. In a CSV HR feed, what is the definition of the name attribute?
 - a. The attribute that uniquely identifies the person
 - b. The attribute that contains the full name of the person
 - c. The attribute that is used by IBM Tivoli Identity Manager to resolve account ownerships during reconciliations
 - d. The attribute that contains the fully qualified DN of the person in the IBM Tivoli Identity Manager ou=person container
5. Which methodology can be used to extend the standard password rules?
 - a. None; password rules cannot be extended.
 - b. Password rules can be extended using JavaScript.
 - c. Password Java APIs can be used to extend password rules.

- d. Password rules can be extended using the Pluggable Authentication Module (PAM) framework.
6. A customer has requested a number of changes to the IBM Tivoli Identity Manager operational workflows to support approvals when an IBM Tivoli Identity Manager entity is created. The changes will require the addition of Approval nodes and scripts to the workflow. Which option would be an accurate description of the customization analysis?
- a. Customization of the operational workflows is not allowed.
 - b. The changes will require a JavaScript extension and therefore would be considered customizations.
 - c. Modifications to workflows using standard workflow components are NOT considered customizations.
 - d. Modifications to operational workflows change the basic IBM Tivoli Identity Manager behavior and are therefore considered customizations.
7. Which option describes best practices for scheduling recertification in large organizations?
- a. Schedule on a rolling basis.
 - b. Schedule all accounts for the end of the calendar year.
 - c. Schedule all accounts for the beginning of the calendar year.
 - d. Divide the accounts into quarters and schedule them on a quarterly basis.
8. Which option describes a best practice for improving reconciliation performance?
- a. Only reconcile once per week.
 - b. Disable policy evaluation during all reconciliations.
 - c. Exclude any attributes in the reconciliation that are not related to policy evaluation.
 - d. Increase the size of the memory in the WebSphere JVM (Java Virtual Machine) until the reconciliation meets performance requirements.
9. Which statement best describes the post office function of IBM Tivoli Identity Manager?
- a. It is used to configure which e-mail server is used to send notifications.
 - b. It collects similar notifications for a period of time to combine multiple e-mails into one.

- c. It configures who will receive e-mail notifications for various manual workflow elements and action items.
 - d. It processes e-mail notifications sent to the Tivoli Identity Manager manager account to complete manual workflows automatically.
10. Reconciliations are resource-intensive operations that can take a long time for services with a large account population. Which option will improve reconciliation performance?
- a. Enable IBM Tivoli Identity Manager server-side sorting.
 - b. Limit the number of attributes returned by the adapter and processed by IBM Tivoli Identity Manager.
 - c. Decrease the default maximum duration as specified in the reconciliation schedule.
 - d. Decrease the SearchALUnusedTimeout configuration parameter in the RMI Dispatcher.

Answer key

1. D
2. B
3. CE
4. A
5. C
6. C
7. A
8. C
9. B
10. B

Archived

Archived

Definitions of path variables

Table B-1 contains the default definitions that are used in this guide to represent the HOME directory level for various product installation paths. You can customize the installation directory and HOME directory for your specific implementation. If this is the case, you need to make the appropriate substitution for the definition of each variable represented in this table.

These operating systems use the following value pairs:

- ▶ Microsoft Windows: drive:\Program Files
- ▶ IBM AIX 5L: /usr
- ▶ Other UNIX: /opt

Table B-1 Default definitions for path variables

Path variable	Default definition	Description
DB_INSTANCE_HOME	<ul style="list-style-type: none"> ▶ Windows: path\IBM\SQLLIB ▶ UNIX: AIX 5L, Linux: /home/dbinstance name Sun Solaris: /export/home/dbinstance name 	The directory that contains the database for Tivoli Identity Manager
LDAP_HOME	<p>For IBM Tivoli Directory Server V5.2</p> <ul style="list-style-type: none"> ▶ Windows: path\IBM\LDAP ▶ UNIX: AIX 5L, Linux: path/ldap Solaris: path/IBMldapsv <p>For IBM Tivoli Directory Server V6.0</p> <ul style="list-style-type: none"> ▶ Windows: path\IBM\LDAP\V6.0 ▶ UNIX: AIX 5L, Solaris: path/IBM/ldap/V6.0 Linux: opt/ibm/ldap/V6.0 <p>For Sun Java System Directory Server (formerly Sun ONE Directory Server)</p> <ul style="list-style-type: none"> ▶ Windows: path\Sun\MPS ▶ UNIX: /var/Sun/mps 	The directory that contains the directory server code
IDS_instance_HOME	<p>For IBM Tivoli Directory Server V6.0</p> <ul style="list-style-type: none"> ▶ Windows: drive\ ibmslapd-instance_owner_name For example, the log file might be C:\idsslapd-ldapdb2\logs\ibmslapd.log. ▶ UNIX: INSTANCE_HOME/idsslapd-instance_name For Linux and AIX 5L systems, the default home directory is the /home/instance_owner_name. On Solaris systems, for example, the directory is the /export/home/ldapdb2/idsslapd-ldapdb2 directory. 	The directory that contains the IBM Tivoli Directory Server Version 6.0 instance

Path variable	Default definition	Description
HTTP_HOME	<ul style="list-style-type: none"> ▶ Windows: path\IBMHttpServer ▶ UNIX: path/IBMHttpServer 	The directory that contains the IBM HTTP Server code
ITIM_HOME	<ul style="list-style-type: none"> ▶ Windows: path\IBM\itim ▶ UNIX: path/IBM/itim 	The base directory that contains the Tivoli Identity Manager code, configuration, and documentation
WAS_HOME	<ul style="list-style-type: none"> ▶ Windows: path\WebSphere\AppServer ▶ UNIX: path/WebSphere/AppServer 	The WebSphere Application Server home directory
WAS_NDM_HOME	<ul style="list-style-type: none"> ▶ Windows: path\WebSphere\DeploymentManager ▶ UNIX: path/WebSphere/DeploymentManager 	The home directory on the deployment manager
Tivoli_Common_Directory	<ul style="list-style-type: none"> ▶ Windows: path\ibm\tivoli\common\CTGIM ▶ UNIX: path/ibm/tivoli/common/CTGIM 	The central location for all serviceability-related files, such as logs and first-failure capture data

Archived

Related publications

We consider the publications that we list in this section particularly suitable for a more detailed discussion of the topics that we cover in this book.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 228. Note that some of the documents that we reference here might be available in softcopy only.

- ▶ *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014
- ▶ *Integrated Identity Management using IBM Tivoli Security Solutions*, SG24-6054
- ▶ *Identity Management Design Guide using IBM Tivoli Identity Manager*, SG24-6996
- ▶ *Identity Management Advanced Design for IBM Tivoli Identity Manager*, SG24-7242
- ▶ *Deployment Guide Series: IBM Tivoli Identity Manager 5.0*, SG24-6477

Other publications

These publications are also relevant as further information sources:

- ▶ IBM Tivoli Identity Manager Information Center
<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>
- ▶ *IBM Tivoli Identity Manager Server Version 5.0 Installation and Configuration Guide*, SC32-1562
- ▶ *IBM Tivoli Identity Manager Version 5.0 Database and Schema Reference*, SC32-9011
- ▶ *IBM Tivoli Identity Manager Version 5.0 Problem Determination Guide*, SC32-1561

Technical supplement

- ▶ *IBM Tivoli Identity Manager Version 5.0 Performance Tuning Guide, SC32-6594*

Online resources

The following Web sites are also relevant as further information sources:

- ▶ IBM Tivoli Software Training
<http://www.ibm.com/software/tivoli/education>
- ▶ IBM Tivoli Identity Manager software information center
<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>
- ▶ Professional Certification Program from IBM
<http://www.ibm.com/certify>

How to get IBM Redbooks publications

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

- access
 - entitlement 26, 44
 - recertification 126
 - provisioning model 60
 - request workflow 55, 129
 - rights 47
- access control
 - for resources 42
 - management 40
- access control item, *see* ACI
- access.log 186, 188
- account 44
 - creation 107
 - for Tivoli Identity Manager 109
 - defaults 127
 - form 94
 - inactivity 131
 - recertification 13, 126
 - request workflow 55, 128
 - self-service 136
 - service type 108
 - Tivoli Identity Manager manager 137
- accountability 53
- ACI 16, 42, 51, 138
 - creation 16
 - scopes 140
 - system administrators 52
- activity
 - workflow 131
- adapter 43, 46, 72, 81
 - ADK-based 97
 - design 12
 - hardware requirements 73
 - installation 15, 89, 93
 - migration 214
 - profile 15, 93, 107
 - project plan 12
 - RMI-based 95
 - service connection 107
 - SSL configuration 97, 99
 - Tivoli Directory Integrator based 109
 - troubleshooting 191

- upgrades 25
- Adapter Development Toolkit 96
- adhocreporting.properties 156
- admin domain 42, 104
- administration 42
 - domain 16
 - policy automation 61
- administrative console 63
 - customization 161
 - view 141
- administrator 137
 - group 137
- adoption policy 112
 - alias 177
 - for identity feed 177
- adoption rule 22
- agent-based adapter 81
- agentCfg 97, 99
- agentless adapters 81
- alias 177
- approval
 - workflow 55
- archival 71
- attribute mapping file 176
- audit 40, 71, 75
 - compliance 77
 - log 192
- auditing 142
- auditor
 - group 137
 - reporting ACI 144
- availability 13, 40, 70

B

- backup 24, 71
 - strategy 13
- business
 - process 10, 13
 - requirements 76
- business partner organization 16, 42, 104
- business partner person 44, 106

C

- certificate 84
 - format 85
 - signing request 15
- certification
 - benefits 3
 - checklist 5
 - IBM Professional Certification 2
- Certified Deployment Professional 7
- challenge/response 119
 - function 45
- cipher suite 86
- comma-separated value, *see* CSV
- compliance 71, 122
 - requirements 26
- configuration
 - adapter 93
- conflict resolution 198
- contract expiration 131
- correct non-compliance 112
- crystal.properties 156
- CSV
 - identity feed 109, 174
- custom
 - adapter 73, 77
 - design 12
 - group 137
 - Java class service 110
 - person 107, 173
- customization 62
 - administrative console 63, 161
 - design 12
 - GUI 62
 - self-service user interface 65, 163
- CustomLabels.properties 64, 156, 162

D

- DAML 99
- data
 - management 21
 - migration 196
 - protection 14
- database
 - troubleshooting 180
- DataBaseFunctions.conf 156
- DB2
 - installation 89
- default escalation period 136

- delegated administration 42
- delivery strategy
 - identity management 73
- design report 144
- dictionary
 - for passwords 118
- difference evaluation 198
- directory information tree 165
- directory server 165
 - configuration 16
 - installation 89
 - object classes 173
 - troubleshooting 181
- Discretionary Access Control 59
- distinguished name 165
- domain
 - administration 43
- DSML
 - identity feed 108, 174
 - placement rule 176
- dynamic organizational roles 105

E

- educational resources 27
- e-mail
 - management 10, 50
 - notifications 150
- enforcement
 - provisioning policy 123
 - rule 123
- enRole.properties 153
- enRoleAuditing.properties 143, 156
- enRoleAuthentication.properties 156
- enRoleDatabase.properties 156
- enRoleLDAPConnection.properties 85, 157
- enRoleLogging.properties 157
- enRoleMail.properties 157
- enRolepolicies.properties 157
- enRoleworkflow.properties 157
- entitlement 120
 - attributes 122
 - automation 61
 - default attributes 128
 - target type 121
 - workflow 128
- entity 44, 129
 - creating 17
 - design 10

- entity type
 - ACI 139
- escalation 55
 - participant 135
 - period 136
- event notification
 - identity feed 174

F

- fesixtensions.properties 157, 214
- fix pack
 - installation 25
- form
 - customization 17, 64
 - designer 162
- functional
 - dependency 197, 199
 - requirements 40

G

- group 51, 136
 - custom 137
 - membership 46
- GUI
 - customization 62

H

- handshake 86
- help desk
 - group 137
- helpmapping.properties 63, 161
- helpmappings.properties 158
- high availability 70
- hosted service 109
- HR
 - data 10
 - feed 21, 75, 81
 - repository 172

I

- IBM Certified Deployment Professional 7
- IBM Key Management utility 86
- IBM Professional Certification 2
- IBM Software Support 193
- IBM Tivoli Access Manager, *see* Tivoli Access Manager
- IBM Tivoli Directory Integrator, *see* Tivoli Directory

Integrator

- IBM Tivoli Identity Manager, *see* Tivoli Identity Manager
- IBM Tivoli Software Professional Certification 4
- identity 47
 - data sources 21
 - design 11
- identity feed 54, 172
 - adoption policy 177
 - for Tivoli Active Directory 108, 175
 - service type 108
 - using CSV file 109, 174
 - using DSML 108, 174
 - using iNetOrgPerson 109, 175
 - using Tivoli Directory Integrator 109, 175
- identity management
 - delivery strategy 73
- identity policy 19, 57, 103, 113, 197
 - design 11
- IDI data feed 109, 176
 - placement rule 176
- implementation 16
- Import/Export 196
- iNetOrgPerson 44, 107, 173
 - identity feed 109
- initial solution rollout project plan 12
- installation 14
 - adapter 15, 93
 - error troubleshooting 180
 - log files 182
 - process 88
- interfaces 40

J

- Java
 - custom class service 110
 - properties 153
- JavaScript
 - extension 26, 214
 - workflow 135
- join directive 19, 122, 124

K

- key format 85

L

- LDAP

- attributes 11
- custom attributes 17
- iNetOrgPerson 44
- object classes 173
- organizationalPerson 44
- schema changes 44
- life cycle
 - management 10, 47, 130
 - rule 49, 197
- location 16, 41, 104
- log files 22, 182
- logical dependency 197, 199

M

- maintenance 24
- managed object 129
- manager
 - group 137
 - reporting ACI 144
- Mandatory Access Control 59
- manual service 110
 - password requirements 45
 - reconciliation 202
- mark non-compliance 112
- membership 120
- message
 - log 187
- migration 196
- modification cycle 48
- monitoring 24, 40
- msg.log 186
- mutual SSL 87

N

- notification
 - e-mails 50
 - template 50
 - workflow 136

O

- objectives
 - for Test 934 9
 - planning 9
- operation workflow 49, 129
- organization
 - identity feed 172
 - structure 9, 16

- tree 41, 44, 103, 172, 199
- tree elements 104
- organizational
 - requirements 11
 - role 42, 104
 - creation 17
- organizationalPerson 44, 173
- orphan account 22, 45, 53–54, 74, 110, 112, 177, 200

P

- page size 20
- password
 - challenge/response 119
 - change frequency 49
 - custom generator 117
 - dictionary 118
 - expiration 131
 - management 40, 74
 - notification method 20
 - self-service 58
 - settings 19
 - strength 42, 58
 - synchronization 19, 45, 117–118
- password policy 19, 42, 57, 117
 - design 11
- performance
 - monitoring 205
 - tuning 25
- person 44, 106
 - attributes 172
 - custom 107
- PKCS12 86
- placement rule 21, 104–105, 176
- planning 9
- policy 42, 74, 113
 - checking 111
 - enforcement 112, 123
 - enforcement type 18
 - management 47
 - preview 23
 - violation 200
- post office 20, 51, 148
 - e-mail notifications 150
- prerequisites 8
- privacy 47
- private key 86
- process

- workflow 131
- production 23
 - environment 195
- project
 - planning 73
 - scope 40
- provisioning 42, 75
 - cycle 48
 - workflow 55
- provisioning policy 18, 53–54, 104, 107, 119
 - access control 42
 - account default 127
 - configuration 157
 - design 10
 - enforcement 121, 123
 - entitlement 120
 - for identity feeds 177
 - join directive 122, 124
 - preview 121
 - simulation 121
 - workflow 120, 135

R

- random password 45
- recertification 13
 - policy 26, 49, 126
 - report 49
- reconciliation 22, 45, 73, 110, 199
 - adoption policy 112
 - identity feed 174
 - manual service 202
 - policy checking 111
 - schedule 23
- recovery 40
 - strategy 13
- recycle bin 203
- Redbooks Web site 228
 - Contact us xii
- registration cycle 48
- reporting 40, 143
 - recertification activities 49
 - requirements 13
- reportingLabels.properties 158
- reporttabledeny.properties 158
- request-based
 - access provisioning model 60
- resource
 - access control 42

- entitlements 61
- RMI
 - adapter 81
 - based adapter 95
 - Dispatcher 81, 95
- role 42, 74, 76, 104
 - creation 17
 - design 10
- Role Based Access Control 59
- role-based
 - access provisioning model 60
- role-based access control 61, 75–76
- runConfig 91, 151, 203

S

- scalability 13, 40
- schema
 - changes 44
 - information 94
- scope 40, 103, 125, 140
- scriptframework.properties 158, 214
- Secure Sockets Layer, *see* SSL
- security
 - audit compliance 77
 - compliance 122
 - model 59
 - model design 11
 - policy 40
- self-registration 176
- self-service
 - account 136
 - ACI 138
 - customization 21
 - password management 58
 - user interface 14, 52, 65
 - customization 163
 - view 141
 - Web layout 67
- SelfServiceHelp.properties 66, 158, 164
- SelfServiceHomePage.properties 66, 158, 164
- SelfServiceScreenText.properties 66, 158, 164
- SelfServiceScreenText_language.properties 66, 164
- SelfServiceScreenTextKeys.properties 66, 164
- SelfServiceUI.properties 66–67, 159, 163
- self-signed certificate 84
- server
 - installation 90

- service 43–44, 107
 - access control 42
 - definition
 - configuration 93
 - form 94
 - owner
 - dashboard 141
 - group 137
 - reporting ACI 144
 - profile 43, 73, 196
 - installation 199
 - reconciliation 110
 - type 18, 107
 - password policy 118
- service selection policy 18, 55, 125
- serviceability tool 193
- single scope 125
- sponsor 107
- SSL 83
 - for ADK-based adapters 99
 - for RMI-based adapters 97
 - handshake 86
 - keystore 84
 - truststore 84
- static organizational roles 105
- static role 103
 - creation 17
- sub-tree scope 125
- suspend non-compliance 112
- system administrator 137
- system architecture 69
 - document 12
- system properties
 - configuration 151
 - enRole.properties 153
- system user 137

T

- target
 - audience 7
 - system 40
 - type 121
- technical requirements 40
- termination 49
- termination phase 49
- Test 934
 - objectives 9
- test strategy 14

- Tivoli Access Manager
 - WebSEAL 45
- Tivoli Active Directory
 - identity feed 108, 175
- Tivoli Common Reporting Server 27
- Tivoli Directory Integrator 73, 81
 - based adapters 109
 - HR feed 81
 - identity feed 175
 - IDI data feed 109
 - installation 16, 89
- Tivoli Identity Manager
 - configuration
 - server installation 90
 - service 109
 - V5.0 enhancements 26
- Tivoli Identity Manager manager
 - default account 137
- Tivoli Software Professional Certification 4
- To Do item 135
- trace data 189
- trace.log 186
- training information 28
- transition
 - workflow 134
- transport
 - protection 14
- troubleshooting 22, 180
 - adapter 191
- tuning 25, 205
- two-way SSL 87

U

- ui.properties 20, 63, 159, 161
- upgrade 24, 210
 - planning 13
- user
 - administration
 - policy automation 61
 - interface 14
 - customization 160
 - parameters 20
 - view 141
 - management 40
 - type 106

V

- view 52, 141

W

- WebSEAL 45
- WebSphere Application Server
 - configuration 16
 - installation 90
 - log files 183
 - SSL keystore and truststore 84
 - troubleshooting 181
- workflow 55, 75, 120, 128
 - access request 129
 - account request 128
 - activity 131
 - activity participants 134
 - automation 56
 - configuration 157
 - creation 18
 - design 11, 55
 - elements 131
 - e-mail notifications 150
 - escalation participant 135
 - escalation period 136
 - for identity feeds 177
 - JavaScript 135
 - notification 20, 136
 - operation 49
 - participant 135
 - process 131
 - provisioning 55
 - recertification 126
 - recertification policy 50
 - transition 134

X

- XHTML 50

Archived



Certification Study Guide: IBM Tivoli Identity Manager Version 5.0

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



Certification Study Guide: IBM Tivoli Identity Manager Version 5.0



Developed specifically for IBM Tivoli Identity Manager certification

This IBM Redbooks publication is a study guide for the “IBM Certified Deployment Professional - IBM Tivoli Identity Manager V5.0” certification test, test number 934, and is meant for those who want to achieve IBM Certifications for this specific product.

Explains the certification path and prerequisites

The IBM Certified Deployment Professional - IBM Tivoli Identity Manager V5.0 certification, offered through the Professional Certification Program from IBM, is designed to validate the skills required of technical professionals who work in the implementation of the IBM Tivoli Identity Manager Version 5.0 product.

Includes sample test questions and answers

This book provides a combination of theory and practical experience needed for a general understanding of the subject matter. It also provides sample questions that will help in the evaluation of personal progress and provide familiarity with the types of questions that will be encountered in the exam.

This book does not replace practical experience, and it is not designed to be a stand-alone guide for any subject. Instead, it is an effective tool that, when combined with education activities and experience, can be a very useful preparation guide for the exam.

**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**