

Identity Management Design Guide with IBM Tivoli Identity Manager

Enterprise integration for identity life
cycle management

Complete architecture and
component discussion

IBM Tivoli Access
Manager integration



Axel Buecker
Dr. Werner Filip
Jaime Cordoba Palacios
Andy Parker



International Technical Support Organization

**Identity Management Design Guide with IBM Tivoli
Identity Manager**

November 2009

Archived

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

Fourth Edition (November 2009)

This edition applies to Version 5, Release 1 of IBM Tivoli Identity Manager.

© Copyright International Business Machines Corporation 2003, 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	xi
Trademarks	xii
Preface	xiii
The team who wrote this IBM Redbooks publication	xiii
Become a published author	xv
Comments welcome	xvi
Part 1. Architecture and design	1
Chapter 1. Business context for identity and credential management ...	3
1.1 Security policies, risk, due care, and due diligence	6
1.2 Centralized user management	8
1.2.1 Single interface	8
1.2.2 Security policy enforcement	8
1.2.3 Central password management	9
1.2.4 Delegation of administration	9
1.2.5 User self-care	9
1.2.6 Multiple repository support	10
1.2.7 Workflow	11
1.2.8 Centralized auditing and reporting	11
1.3 Simplify user management	13
1.3.1 Automation of business processes	13
1.3.2 Automated default and validation policies	14
1.3.3 Single access control models	14
1.3.4 Ubiquitous management interfaces	14
1.3.5 Integration of other management architectures	15
1.4 Centralized group management	15
1.4.1 Group management	15
1.4.2 Compliance	16
1.5 Life cycle management	16
1.5.1 Recertification policies	17
1.5.2 Life cycle rules	19
1.6 Access control models	20
1.6.1 The role-based access control model	20
1.6.2 Other access control models	21
1.6.3 Which model	22
1.6.4 Selection process	24
1.6.5 Roles versus groups	29

1.6.6 Designs	30
1.6.7 Observations	35
1.7 Identity management versus meta directory	35
1.8 Identity management provisioning models	43
1.8.1 Request-based approach	44
1.8.2 Role-based approach	44
1.8.3 Hybrid approach	45
1.9 Role management	45
1.9.1 Relationship and hierarchy	45
1.10 Separation of duties	47
1.11 Conclusion	47
Chapter 2. Architecting identity and credential management solutions	51
2.1 Solution architectures, design, and methodologies	52
2.1.1 Overview and terminology	52
2.1.2 Implementation flow	54
2.1.3 Definition of an identity management solution	58
2.1.4 Design of an identity management solution	60
2.1.5 Data model considerations	61
2.1.6 Migration considerations	64
2.2 Identity management design process	66
2.2.1 MASS and identity management	67
2.2.2 Developing security architectures using MASS	72
2.2.3 Integration into the overall solution architecture	74
2.3 Business processes and identity management	80
2.4 Conclusions	82
Chapter 3. Tivoli Identity Manager component structure	83
3.1 Logical component architecture	84
3.1.1 Application layer	85
3.1.2 Service layer	88
3.1.3 LDAP Directory	91
3.1.4 Database	92
3.1.5 Resource connectivity	92
3.2 Physical component architecture	94
3.2.1 Component configuration and placement	94
3.2.2 Network zones	94
3.2.3 Integrating with Tivoli Access Manager	97
Chapter 4. Detailed component design	101
4.1 Tivoli Identity Manager entities	102
4.1.1 Users, accounts, services, and attributes	102
4.1.2 Passwords	103
4.1.3 Group membership	105

4.1.4	Accesses	106
4.1.5	Managed systems and applications	108
4.2	Tivoli Identity Manager management entities	110
4.2.1	Organizational tree	111
4.2.2	Organizational roles	113
4.2.3	Tivoli Identity Manager groups and ACIs	116
4.2.4	Policy.	117
4.2.5	Account defaults	121
4.2.6	Workflow	122
4.2.7	Logs and audit.	123
4.2.8	Reports	124
4.2.9	Entity relationships	127
4.2.10	Life cycle management	129
4.3	Tivoli Identity Manager functions.	130
4.3.1	Tivoli Identity Manager configuration and user self-care interfaces	131
4.3.2	Password management	134
4.3.3	Manage people and accounts.	138
4.3.4	Apply policies to person and account management	141
4.3.5	Reconcile accounts.	158
4.3.6	Apply workflow to people and account management	161
4.3.7	Produce reports.	170
4.3.8	E-mail notification	173
4.3.9	Manage activities	177
4.3.10	Import/export.	181
4.3.11	Managing groups	185
4.4	User interface and access control.	187
4.5	Tivoli Identity Manager schedules.	196
4.5.1	Scheduling of changes	197
4.5.2	Scheduling of reconciliations.	199
4.5.3	Time limits on workflow	200
4.5.4	Recertification policies	202
4.5.5	Life cycle rules	203
4.5.6	Post office	204
4.5.7	Historical reporting	206
4.6	Common customization.	208
4.6.1	JavaScript extensions	208
4.6.2	Application clients	209
4.6.3	Workflow definitions	210
4.6.4	Workflow application extensions.	210
4.6.5	Password rules	211
4.6.6	Data services API	211
4.6.7	Identity feeds.	212
4.6.8	Custom person classes.	213

4.6.9 Custom adapters	213
4.6.10 User interface	214
4.6.11 Subforms	215
4.7 Adapter connectivity	216
4.8 Software and hardware requirements	219
4.8.1 Software requirements	219
4.8.2 Hardware requirements	220
Chapter 5. Operational solution design	223
5.1 Maintainability and configuration management	227
5.1.1 Version control	227
5.1.2 Multiple environments	229
5.1.3 Migration between environments	232
5.1.4 Managing system accounts	235
5.2 Archival and backup	236
5.2.1 Archival	237
5.2.2 Backup	240
5.3 High availability and failure recovery	241
5.3.1 Application server	242
5.3.2 Java Messaging Service	243
5.3.3 Directory server	249
5.3.4 Relational database	254
5.3.5 Tivoli Identity Manager adapters	256
5.4 Monitoring	260
5.4.1 Tivoli Identity Manager infrastructure	262
5.4.2 Tivoli Identity Manager application	262
5.4.3 Sample monitoring tools and products	266
5.5 Security and integrity	271
5.6 Conclusion	273
Chapter 6. Tivoli Access Manager integration	275
6.1 Functional overview	276
6.2 Managing Tivoli Access Manager with Tivoli Identity Manager	278
6.2.1 Creating a Tivoli Access Manager service	278
6.2.2 Setting up a Tivoli Access Manager specific policy	279
6.2.3 Managing group memberships for Tivoli Access Manager	286
6.2.4 Reconciliation	286
6.2.5 Automatic provisioning	288
6.2.6 Delegated user administration	288
6.3 Resource versus user management	290
6.3.1 Different types of administrators	290
6.3.2 Resource management	291
6.4 Integration with Tivoli Access Manager WebSEAL	291

6.4.1	Managing LDAP attributes used by WebSEAL	292
6.4.2	Setting up Web single sign-on	294
6.5	Tivoli Access Manager for Operating Systems	295
Part 2.	Customer environment	297
Chapter 7.	Tivoli Austin Airlines, Inc.	299
7.1	Company profile	300
7.1.1	Geographic distribution of TAA	300
7.1.2	Organization of TAA	304
7.1.3	HR and personnel procedures	305
7.2	Current IT architecture	306
7.2.1	Overview of the TAA network	306
7.2.2	TAA's e-business initiative	308
7.2.3	Security infrastructure for the e-business initiative	309
7.2.4	Secured e-business initiative architecture	311
7.2.5	Identity management and emerging problems	314
7.3	Corporate business vision and objectives	316
7.4	Project layout and implementation phases	316
7.5	Return on investment (ROI) study and results	317
Chapter 8.	Identity management design	323
8.1	Business requirements	324
8.2	Functional requirements	325
8.3	Design approach	334
8.4	Implementation approach	335
Chapter 9.	Technical implementation: Phase I	341
9.1	Initial installation and configuration	343
9.1.1	Requirements	343
9.1.2	Design considerations	343
9.1.3	TAA's implementation	344
9.2	Secure the Tivoli Identity Manager application	349
9.2.1	Requirements	349
9.2.2	Design considerations	349
9.2.3	TAA's implementation	353
9.3	Organization tree	355
9.3.1	Requirements	355
9.3.2	Design considerations	355
9.3.3	TAA's implementation	356
9.4	Services	358
9.4.1	Requirements	358
9.4.2	Design considerations	358
9.4.3	TAA's implementation	360

9.5 Provisioning policies	363
9.5.1 Requirements	363
9.5.2 Design considerations	364
9.5.3 TAA's implementation	364
9.6 Load users into Tivoli Identity Manager	368
9.6.1 Requirements	368
9.6.2 Design considerations	368
9.6.3 TAA's implementation	380
9.7 Reconciliation	404
9.7.1 Requirements	405
9.7.2 Design considerations	405
9.7.3 TAA's implementation	407
9.8 Orphan account cleanup	414
9.8.1 Requirements	414
9.8.2 Design considerations	414
9.8.3 TAA's implementation	416
Chapter 10. Technical implementation: Phase II	419
10.1 Common account creation	421
10.1.1 Requirements	421
10.1.2 Design considerations	421
10.1.3 TAA's implementation	426
10.2 Password policy	431
10.2.1 Requirements	431
10.2.2 Design considerations	432
10.2.3 TAA's implementation	432
10.3 Password strength policy	433
10.3.1 Requirements	433
10.3.2 Design considerations	434
10.3.3 TAA's implementation	434
10.4 Password synchronization using the Windows password interceptor	436
10.4.1 Requirements	437
10.4.2 Design considerations	437
10.4.3 TAA's implementation	438
10.5 Account suspension on termination	443
10.5.1 Requirements	444
10.5.2 Design considerations	444
10.5.3 TAA's implementation	445
10.6 Reporting considerations	448
Chapter 11. Technical implementation: Phase III	449
11.1 Password challenge/response	452
11.1.1 Requirements	452

11.1.2	Design considerations	452
11.1.3	TAA's implementation	452
11.2	Account management using the Web user interface	456
11.2.1	Requirements	456
11.2.2	Design considerations	456
11.2.3	TAA's implementation	461
11.3	Delegation	461
11.3.1	Requirements	461
11.3.2	Design considerations	462
11.3.3	TAA's implementation	478
11.4	Tivoli Identity Manager change control	503
11.4.1	Requirements	503
11.4.2	Design considerations	504
11.4.3	TAA's implementation	507
Chapter 12. Technical implementation: Phase IV		511
12.1	Preparing for role-based access control	512
12.1.1	Knowledge gathering approaches	514
12.1.2	Setting expectations	519
12.1.3	Defining roles	521
12.1.4	Defining provisioning policies	528
12.1.5	Defining accesses	535
12.1.6	Defining separation of duty policies	536
12.2	Requirements	538
12.3	Design considerations	539
12.4	TAA implementation	540
12.4.1	Location codes	540
12.4.2	Employee codes	541
12.4.3	Roles	544
12.4.4	Provisioning policies	562
12.4.5	Separation of duty policy	566
12.4.6	Accesses	573
12.5	Certification process	578
12.5.1	Requirements	578
12.5.2	Design considerations	579
12.5.3	TAA's critical account recertification implementation	579
12.5.4	TAA's role and group recertification implementation	598
12.5.5	Responding to recertification activities	603
12.6	Future deployment phases	606
12.6.1	Advanced reporting	606
12.6.2	Advanced workflow customization	607
12.6.3	Adapter customization	608
12.6.4	User interface graphics update	609

12.6.5 Additional examples	610
12.7 Conclusion	610
Appendix A. Account management workflow customization	615
Requirements	616
Design considerations	616
TAA's implementation	620
Conclusion	629
Appendix B. Windows desktop password reset and unlock	631
Requirements	632
Design considerations	633
TAA's implementation	633
DPRA installation	634
Using DPRA	637
DPRA customization	639
Conclusion	641
Appendix C. Automating tasks for role management	643
Requirements	644
Design considerations	644
TAA's implementation	644
Installation and configuration of Apiscript	645
Bulk loading static organizational roles	646
Populating static organizational roles using apiscript	649
Conclusion	654
Appendix D. Additional material	655
Locating the Web material	655
Using the Web material	656
System requirements for downloading the Web material	656
How to use the Web material	656
Glossary	657
Related publications	665
IBM Redbooks publications	665
Other publications	666
Online resources	667
How to get IBM Redbooks publications	667
Index	669

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®	IBM®	Redbooks®
alphaWorks®	Lotus Notes®	Redpaper™
DB2 Universal Database™	Lotus®	Redbooks (logo)  ®
DB2®	Notes®	System x®
developerWorks®	Passport Advantage®	Tivoli®
HACMP™	RACF®	WebSphere®
i5/OS®	RDN®	z/OS®

The following terms are trademarks of other companies:

Adobe, the Adobe logo, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

Identity management has always been an important factor in the systems management discipline. Managing individual users with all their different accounts on multiple platforms and applications in a timely manner is a crucial Internet technology (IT) task. In the emerging world of e-business and worldwide Internet connectivity, this discipline is becoming a major part of an enterprise security architecture.

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 it 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 an approach for designing an identity management solution with IBM Tivoli® Identity Manager Version 5.1. Starting from the high-level, organizational viewpoint, we show how to define user registration and maintenance processes using the self-registration and self-care interfaces as well as the delegated administration capabilities. Using the integrated workflow, we automate the submission/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 wish to understand and implement a centralized identity management and security infrastructure.

In this edition, we have updated the overall content to cover the new version of Tivoli Identity Manager.

The team who wrote this IBM Redbooks publication

This IBM Redbooks publication was produced by a team of specialists from around the world working at the International Technical Support Organization, 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 23 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.



Dr. Werner Filip is a professor of the department for Computer Science and Engineering at the University of Applied Sciences Frankfurt am Main, Germany and a Consultant in IT Security. His primary research interests are Systems and Network Management and Applied Security. Prior to joining University of Applied Sciences Frankfurt, he worked for 25 years for IBM in various positions, and during his last 10 years with IBM, as a Consultant in Systems and Network Management at the former IBM European Networking Center, Germany. He received a Diploma in Mathematics, and a Doctorate in Computer Science from the Technical University Darmstadt, Germany.



Jaime Cordoba Palacios is a Certified Consulting IT Specialist with Grafica Consultora (an IBM Business Partner), based in Mexico, D.F. He holds a degree in Electronics Engineering, and a degree in Information Security from Instituto Tecnológico y de Estudios Superiores de Monterrey, Mexico. He has 10 years of experience in a variety of areas related to systems management, network computing, and security solutions. His areas of expertise include IBM Tivoli Access Manager, IBM Tivoli Risk Manager, IBM Tivoli Identity Manager, LDAP, e-business infrastructures, and networking. He currently is involved in security architecture design and implementation and general security consulting engagements.



Andy Parker is an IBM IT Specialist with 20 years of IT experience. He is a Certified Information & System Security Professional (CISSP) working for IBM Software group at UK Hursley Labs. Andy has been working for Software Group since 2000 on various security related projects, the most recent project being the Software group Hursley Labs IBM Tivoli Identity Manager deployment. He joined IBM in 1995 working as the team lead for the communications team in the AIX® support center. Prior to working for IBM he worked for ICL, now part of Fujitsu, on a security project for the UK government. Before working at ICL, he served seven years in the British Army.

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

Wade Wallace
IBM ITSO

Chris Brooks, Dennis Doll, Jeremy Finney, Bassam Hassoun, Dinesh Jain, Sam Montgomery-Blinn, Paul O'Mahoney, David Palmieri, Kung Pao, Casey Peel, Jeffrey Robke, Pat Saunders, Charlie Saylor, Corey Williams, and the whole Tivoli Identity Manager development team
IBM Corporation

Thanks to the authors of the previous editions of this book.

- ▶ Authors of the first edition, *Identity Management Design Guide with IBM Tivoli Identity Manager*, published in July 2003, were:
Andrew Camp, Rick Cohen, David Edwards, Collin Penman, Thomas Santana
- ▶ Authors of the second edition, *Identity Management Design Guide with IBM Tivoli Identity Manager*, published in November 2005, were:
Jaime Cordoba Palacios, Brian Davis, Todd Hastings, Ian Yip
- ▶ Authors of the third edition, *Identity Management Design Guide with IBM Tivoli Identity Manager*, published in April 2009, were:
Andrew Annas, Dale LeFevre, Philippe Lottmann, Pär Kidman, Ralf Willert, Jason Wu

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbooks publication dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners, or clients.

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

To obtain more information 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 Redbooks publications to be as helpful as possible. Send us your comments about this or other Redbooks publications in one of the following ways:

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

ibm.com/redbooks

- ▶ Send your comments in an Internet note to:

redbook@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



Part 1

Architecture and design

In this part, we introduce the general components of IBM Tivoli Identity Manager V5.1 and what it has to offer in the identity and credential management area of the overall security architecture. Tivoli Identity Manager handles a multitude of integration aspects with all sorts of Internet technology (IT) infrastructures, Web portals, and application environments, which are detailed throughout this part of the book. After talking about architectures and design, Part 2, “Customer environment” on page 297, provides a more solution-oriented, scenario-based approach.

Archived

Business context for identity and credential management

As the world of e-business gains global acceptance and access to these systems becomes mission critical, the traditional processes of corporate user administration are no longer able to cope with the demands for increased scale, scope, and availability expected from them. Identity management is a super-set of older user provisioning systems that allows for the management of identity and credential information for clients, partners, suppliers, automated processes, corporate users, and others. New functional capabilities provide businesses with an opportunity to re-engineer their procedures for managing access to their IT resources based on their business policies, which in turn drive their IT security policies and their IT security procedures.

Unfortunately, this attracts attention from both people outside the business, who want to use these Internet technology (IT) assets for illicit purposes, and insiders who may be seeking to benefit from their privileged access to sensitive information.

Legislation is being enacted worldwide to ensure the integrity of a corporation's IT assets, especially those assets that determine the corporation's financial results. New audit and compliance reporting rules are the result. For example, in June 2004, central bank governors and bank supervisory authorities for members of the Group of Ten (G10) countries endorsed the publication of the "International Convergence of Capital Measurement and Capital Standards: a revised framework," commonly called Basel II¹. This product provided financial incentives for banks worldwide to upgrade and improve their business models, their risk management systems, and their public disclosure information to provide greater transparency of their operations. Banks must manage their capital resources efficiently because it not only affects their profitability, capital provides the foundation for growth and the cushion against an unexpected loss.

In the United States, the Sarbanes-Oxley Act² of 2002 requires that all publicly held corporations that are being traded on the United States stock exchange with more than three hundred shareholders to provide information about the accuracy of their financial records and the internal controls to the financial data. This legislation has created a ripple effect in the international community because the Sarbanes-Oxley requirements may exceed legislation in the countries where these international companies have their headquarters. In some cases, the Sarbanes-Oxley requirements may conflict with the local legislation.

Companies that are implementing accounting and audit procedures to comply with the Sarbanes-Oxley legislation are stating that the core problem is identifying who has access to the financial information and the business reasons they have been given this access. It comes down to an identity management and provisioning challenge.

The Gramm-Leach-Bliley Act³ (GLBA) of 1999 established regulations for the protection and privacy of an individual's financial information that is maintained by private organizations. Compliance was mandated by July 2001.

The Payment Card Industry Data Security Standard (PCI DSS)⁴ was developed by the major credit card companies as a guideline to help organizations that process card payments prevent credit card fraud, cracking of various other security vulnerabilities, and threats. A company processing, storing, or transmitting payment card data must be PCI DSS compliant or risk losing their ability to process credit card payments and being audited or fined. Merchants and payment card service providers must validate their compliance periodically.

¹ Find more information about the Basel II framework at <http://www.bis.org/publ/bcbsca.htm>.

² Find more information about the Sarbanes-Oxley Act at <http://www.sarbanes-oxley.com/>.

³ Find more information about the GLBA at <http://www.ftc.gov/privacy/glbaact/>.

⁴ Find more information about the PCI DSS at <http://www.pcisecuritystandards.org/>.

Revisions to existing legislation and new legislation is under consideration to control access to personal information contained in these IT assets such as an individual's health information or financial data. For example, in the United States, the Health Insurance Privacy and Accountability Act⁵ of 1996 (HIPAA) created national standards to protect an individual's medical records and other health information. It gives patients more control over their health records and limits the use of information contained in these records.

The Liberty Alliance Project⁶ has published the first draft of an effort for establishing an Identity Governance Framework. The Identity Governance initiative's objective is intended to help organizations to manage identity information based on policies and demonstrate compliance with the standards and regulations mentioned before, providing visibility, control, auditing and reporting functions.

Organizations must be able to demonstrate due care, due diligence, improved security, and compliance with legislation in all countries where they operate. Unauthorized access has become so pervasive that Data Governance groups are being formed around the world by business consortiums, IT vendors, and telecommunications providers. All these groups share the common goal of developing standards and best practices for securing the data stored on their IT assets.

In this chapter, we talk specifically about the business goals that drive the need for identify management solutions. In addition, we examine some of the concepts surrounding identify management and how they impact costs and business risk mitigation. Life cycle management concepts and the role-based access control Model (RBAC) are examined in greater depth because they are creating a valid return on investment (ROI) and driving better control over the assets of an organization.

⁵ Find more information about HIPAA at <http://www.hhs.gov/ocr/hipaa/>.

⁶ Find more information about the Liberty Alliance Project at http://www.projectliberty.org/strategic_initiatives/identity_governance

1.1 Security policies, risk, due care, and due diligence

The senior management team of an organization must show due care, and in some cases compliance, in all their dealings, including security-related matters. Showing due care or compliance helps to create a professionally managed organization, which in turn helps maintain shareholder value. Due care or compliance can also be an important step towards avoiding claims of negligence resulting from security breaches and reported by the media. From a security perspective, showing due care can be achieved by having well thought out security policies and the appropriate operational controls, processes, and procedures in place to implement the security policy.

Security policies must balance a number of conflicting interests. It is easy to write security policies that deny access or make access controls so onerous that either no business gain can be achieved or the security policies are ignored in order to make reasonable business gains. In some cases, security policies are developed but they are not enforced. Therefore, securities must be monitored, measured, and reported by the IT department according to a schedule that the business defines is commensurate with the importance of the data. Security policies should be audited frequently by a separate organization. There are tools available such as IBM Tivoli Compliance Insight Manager⁷ to automate this effort. However, that is out of scope of this book.

Security policies must set a sensible level of control that takes into account not only the culture and experience of the organization, but an appreciation of the risks involved.

When dealing with regulations the compliance effort can be divided into four areas:

- ▶ Identify data covered by the compliance regulation and define a security policy that satisfies the requirement.
- ▶ Apply operational controls, processes, and procedures to implement the security policy.
- ▶ Validate the ongoing effectiveness of the controls.
- ▶ Demonstrate the process for identifying and managing exceptions to the security policy.

⁷ For more information about Tivoli Compliance Insight Manager refer to the IBM Redbooks publication *Compliance Management Design Guide with IBM Tivoli Compliance Insight Manager*, SG24-7530.

Risk assessment is an important topic in its own right, but is outside the scope of this book. Briefly, risk is usually assessed either formally or informally using quantitative or qualitative methods. It involves assessing what could go wrong, how likely it is to occur, and what damage results from that event. Elements to analyze include threats, probability, damage, and trade-offs.

Risk can be dealt with in one of four ways:

- Transfer risk** The most common way of transferring risk is through insurance. In the current economic environment, the availability and cost of insurance is variable. Currently, this method is more volatile than in the past.
- Mitigate risk** Mitigation of risk can be achieved by identifying and implementing the means to reduce the exposure to risk. This includes the deployment of technologies that improve the security cover within an organization. Deploying an identify management tool mitigates the security risks associated with poor identity management.
- Accept risk** An organization may chose to accept that the impact of the risk is bearable without transferring the risk or mitigating the risk. This is often done where the risk or its impact is small, or when the cost of mitigation is large.
- Ignore risk** Often confused with risk acceptance, ignoring risk is all too common. The main difference between accepting risk and ignoring risk is that risk assessment is an implicit part of risk acceptance. If no valid risk assessment has been done, this should raise a warning flag. This flag points towards the dangerous path of ignoring risk.

Understanding the risks that exist allows us to write appropriate security policies. Having security policies shows the exercise of due care, but unless the policies are implemented, due diligence cannot be shown. Many organizations write good security policies only to fail at the implementation stage, because implementation represents a difficult or costly challenge. In the next section we show how a centralized identity management solution can be used to enforce security policies relating to identity management. This gives us demonstrable due diligence with respect to identity management.

1.2 Centralized user management

The benefits of centralizing the control over user management, while still allowing for decentralized administration, impacts these four business areas:

- ▶ The cost for user management can be significantly reduced.
- ▶ The amount of lost productivity while users wait for their accounts to be created or to have their passwords reset can be reduced.
- ▶ The risks of former employees having access to IT resources after they separate from the business can be reduced.
- ▶ Security policies can be automatically enforced.

Let us take a closer look at the capabilities of centralized user management that help realize these benefits.

1.2.1 Single interface

Most large IT systems today are very complex. They consist of many heterogeneous resources (operating systems, databases, Web application servers, and so on). Individual user accounts exist in every database or user identity repository. This means that an administrator must master a different interface on each platform or resource type in order to manage the user identity repository. This can be compounded by having specialized administrators focusing on specific platforms.

As the number and complexity of operations increases, the result is often an increase in errors due to mistakes, time delays, or coordination problems. This situation can be resolved through the centralization of identity management and implementing role-based access control over the administration of users.

The centralization of the cross-environment management provides a common interface for administration of user identity information, thus reducing education and maintenance costs.

1.2.2 Security policy enforcement

Identity management policies should be implemented as part of the standards and procedures that are derived from the corporate security policy. Implementing identity management policies that comply with the corporate security policy is a key factor for a successful identity and credential management system. Central control makes it possible to accommodate the business and security policies, enabling security administrators to implement them in an efficient and enforceable way.

Without centralized identity management and the use of life cycle rules, it is almost impossible to enforce the corporate policy in a complex environment dealing with a variety of target platforms, different system specifications, and different administrators.

1.2.3 Central password management

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. Password strength policy can also be applied consistently across the enterprise.

Centralized password resets enable a user or administrator to reset one or all account passwords from a central interface. This prevents lost productivity due to the inability to access critical systems.

If a user's password changes on the target resource directly, it may be useful to update the central system in some environments if the password conforms to the password policy or the password change is not allowed. If password synchronization is used, other accounts can be synchronized to maintain consistency.

1.2.4 Delegation of administration

As the number and type of users within the scope of an organization's identity management system changes, there will be increasing burdens on the system. Any centralized system run by an IT department could face the burden of having to manage users who are within other business units or even within other partner organizations.

A key feature of any centralized system is therefore the ability to delegate the day-to-day management of users to nominated leaders in other business units or partner organizations.

The extreme example of delegation is delegation to an individual to manage some features of his own identity. Examples of this would be changing location details or the password self-reset.

1.2.5 User self-care

The most frequent reason users call the Help Desk is because they have forgotten their password and they have locked their account while entering incorrect passwords.

A robust identity management solution should provide users with an automated tool for resetting their passwords based on them supplying correct responses to one or more password challenge questions. Depending on the risks or the classification of data on the server, this tool may send the new password to the user's e-mail address of record or present the user with a Web page to enter a new password on the spot. The tool may also generate audit records and notifications to IT or administrative personnel monitoring user self-care activities.

1.2.6 Multiple repository support

When we talk about repository support, we must look at two types of repositories:

- ▶ User repositories
- ▶ Endpoint repositories

User repositories

User repositories contain data about people, and most companies have many user repositories and will continue to add new ones due to new and custom applications. These can be:

- ▶ Human resources systems
- ▶ Applications
- ▶ Lightweight Directory Access Protocol (LDAP) and other directories
- ▶ Meta directories

Endpoint repositories

Endpoint repositories contain data about privileges and accounts, and most companies have a large variety of these repositories implemented throughout their environment. Some of these are:

- ▶ Operating systems, such as Windows®, AIX, or Linux®
- ▶ E-mail systems such as Microsoft® Exchange or Lotus® Notes®
- ▶ Enterprise Business Applications such as SAP, Oracle e-Business Suite, or Siebel
- ▶ PKI and strong authentication solutions such as Entrust Authority PKI server and RSA Authentication Manager
- ▶ Access Management solutions such as IBM Tivoli Access Manager
- ▶ Network devices
- ▶ Mainframe user repositories such as RACF®

It is important, therefore, when considering centralized identity management systems to be sure that the coverage of the system takes both types of repositories into full account.

1.2.7 Workflow

Managing identity and account-related data involves a great deal of approvals and dependencies. It takes a lot of time and effort to collect the necessary approvals and check for all sorts of dependencies between related components. To reduce these often manually conducted chores, the identity management system should have an automated workflow capability that allows the system to:

- ▶ Gather approvals.
- ▶ Reduce administrative workload.
- ▶ Reduce turn-on time for new managed identities (account generation, provisioning, and so on).
- ▶ Enforce completeness (do not do this task before everything else is gathered).

The workflow component is a core value point within an identity management solution.

1.2.8 Centralized auditing and reporting

Traditionally, many organizations have treated audit logs, on each of the corporate repositories, as places to look for the cause of a security breach after the fact. Increasingly, this will be seen as an inadequate use of the information available to an organization, which could be exhibiting better due diligence by monitoring and reacting to logged breaches in as near real time as possible.

This requirement can only be met using centralized threat management tools, but an important step towards meeting this goal should be part of an identity management solution. Centralized auditing and logging of all additions, changes, and deletions made on target repositories should be part of any centralized identity management solution. See Table 1-1.

Table 1-1 Summary of centralized identity management benefits

Centralized management feature	Cost reduction impact	Security impact
Single interface	Lower skill set required to add, modify, and delete users.	Single interface leads to less human interaction and error.
Security policy enforcement	Because the policy is enforced centrally, less time and cost is spent on enforcement and auditing.	Security risks are reduced because corporate security policies are controlled at the center.
Central password management	Users spend less time managing multiple passwords and productivity gains are therefore made.	Password strength is uniformly applied across the enterprise.
Delegated administration	This allows the organization to offload some of the day-to-day workload and therefore costs.	Changes made to accounts by delegated administrators still must conform to the security policies in force.
Multiple repository support	Including all user repositories in the coverage of identity management solutions reduces cost of specialist administrators.	Including all user and account repositories in the coverage of identity management solutions allows policy to be applied uniformly.
Workflow	Reduced turn-on time and reduced manual administrative operations.	Approval enforcement.
Centralized auditing and reporting	Reduced time spent on audit trails on disparate systems.	Centralized auditing makes the tracing of events more realistic and therefore much more secure.

Considerations

An enterprise identity management solution must provide standard (canned) reports, plus the ability to prepare custom (ad hoc) reports that are designed to address special circumstances. Custom reports may include the modification of a standard report or the creation of a unique report using the audit and log data.

The enterprise identity management solution should have its own client reporting tool and should provide an interface to an external report creation tool. We strongly recommend that all reports be available in more than one format (for example, Adobe® PDF and HTML).

For large enterprises (those in excess of 10,000 accounts), the client reporting tools should have the ability to filter the amount of data reported so that the report may be useful to the person reading it.

Suggested report groups are:

- ▶ Requests reports
- ▶ User and accounts reports
- ▶ Services reports
- ▶ Audit and security reports
- ▶ Custom reports

1.3 Simplify user management

This section discusses how to simplify the user-management process. This is largely achieved by having a clear security policy, a well-organized implementation of the policy, and sensible automation of the necessary processes in place.

1.3.1 Automation of business processes

All user accounts have a life cycle: They are created, modified, and deleted. It can take a long time to get a new user online, as administrators are often forced to manually obtain approvals, provision resources, and issue passwords.

Generally, with manual work, there is the opportunity for human error and *management by mood*. Self-service interfaces enable users to perform some of these operations on their own information, such as password resets and personal information updates.

Automating some of the business processes related to the user account life cycle reduces the chance for error and simplifies operations.

Any centralized identity management solution must provide the means to emulate the manual processes involved in provisioning requests, an approvals workflow, and an audit trail, in addition to the normal provisioning tools.

1.3.2 Automated default and validation policies

When creating user account information, some characteristics are common to all or a sub-set of users based on the context. Default policies, which fill in data entry fields with pre-set values automatically if not specified, reduce the effort to fill out those values for every account.

A validation policy ensures that information about an object complies with the rules defined for that object in the enterprise. An example would be that the field *user name* must be eight characters and start with a letter. Another validation policy may be that every user must have at least one active group membership.

1.3.3 Single access control models

Defining an access control model for each type of resource (e-business, enterprise and previous platforms, and applications) in an organization can be complex and costly. A single access control model provides a consistent way to grant users access to resources and control what access the user has for that resource or across a set of resources.

For some organizations, a role-based access control model is a good goal for which to aim, as this reduces cost and improves the security of identity management. Access control models are discussed in 1.6, “Access control models” on page 20.

1.3.4 Ubiquitous management interfaces

Work styles are changing and not everyone is office bound. Some people may work in a different business location everyday, while others may work from a home office. Identity management interfaces should be ubiquitous to adjust to our work styles. It may be necessary for users in partner organizations or clients to self manage some of their account data. This means that the software on the access device may not be under the control of the parent organization. Since a Web browser interface is a pervasive interface available on most devices, it makes sense that any identity management solution interface should be Web based.

In order for administrators to perform their work tasks anytime from anywhere with a network connection, the identity management solution must be Web

enabled and capable of being integrated with Internet-facing access control systems.

1.3.5 Integration of other management architectures

Identity management is one part of an overall security architecture. Many organizations are experiencing the benefits of automating and centralizing security administration. Integrating identity management with access control solutions and the threat management architecture can help an organization to deploy applications faster and pursue new business initiatives, while enforcing policy compliance across the organization. Security management also must integrate with systems management so that potential threats to an organization can be detected and resolved. For example, if the threat management detects an unpatched application server, operating system, and so on, then the systems management tools should automatically distribute the required patch.

Within the field of identity management, the use of automated provisioning may trigger workflows. Distributing software or updating the configuration of the user's workstation by using the software distribution functionality found in the system's management architecture is one example of the type of functionality required from an identity management solution.

1.4 Centralized group management

In this section, we discuss how group management can be simplified. Similar to the approach mentioned in 1.3, "Simplify user management" on page 13, this is largely achieved by having a clear security policy, a well-organized implementation of the policy, and sensible automation of the necessary processes in place.

1.4.1 Group management

When managing user account information based on policies, it is often necessary to provide group information related to the user. The information related to groups typically changes for every managed system, platform, or application, and in a heterogeneous environment, this can be complex because of the number of systems and the variety of those systems.

A functionality that allows managing groups from a central point enables identity management in a more rapid and efficient way. A user can be granted access to a specific group, or the group can even be created before granting the user access.

Membership in a group typically grants specific access to a resource or a set of resources on a system. When the organization begins to design access control policies, it is very important to consider these capabilities. One capability is the ability to implement a central point for administering those groups, and the other is on which systems it is technically possible to remotely administer group management.

1.4.2 Compliance

Identity management is a part of the overall enterprise security architecture, and it has to comply with regulation efforts within the organization. Since group management can be executed in both a centralized and distributed way, just as with the user management process, a reconciliation function must exist that can help organizations maintain unique information across all of their security architectures.

1.5 Life cycle management

Life cycle management introduces the concept that a person's use of an IT asset from the time that the account is created until the time that the account is deleted will change over time due to external events such as transfers, promotions, leaves of absence, temporary assignments, or management assignments. There may also be a need to routinely verify that the account is compliant with security policies or external regulations. This effort is an on-going process and not a one-time event. Control activities must therefore be implemented into the business processes. Automation increases the effectiveness of these controls and business processes.

Life cycle is a term used to describe how accounts for a person are created, managed, and terminated based on certain events or a time-based paradigm.

Figure 1-1 represents a closed-loop process where a person is registered to use an IT asset, an account is created, and access provisioning occurs to give this person's account access to system resources. Over time modifications occur where access to some resources is granted, while access to other resources may be revoked. The cycle ends when the person separates from the business and the terminate process removes access to resources, suspends all accounts, and eventually deletes the accounts and the person from the systems.

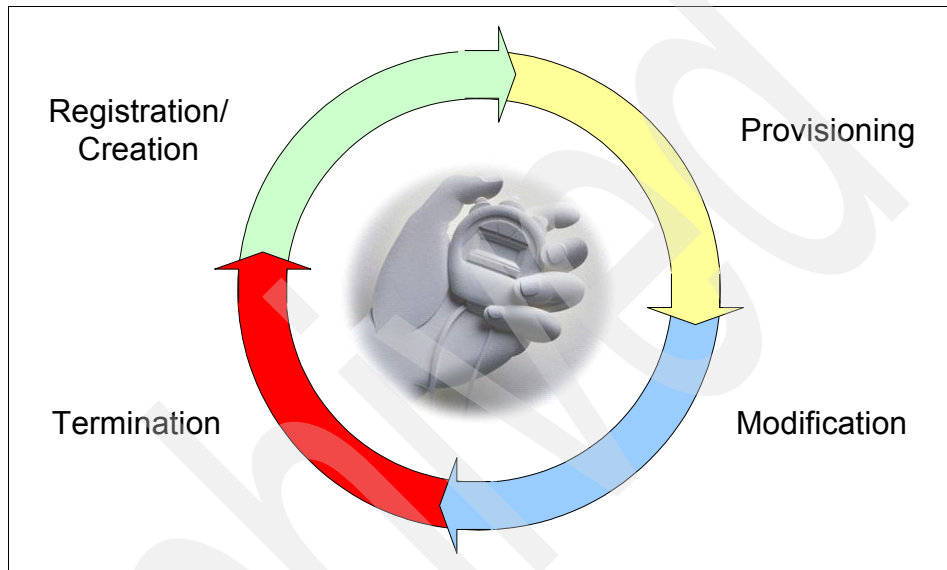


Figure 1-1 Life cycle management overview

There are two ways of automating life cycle management:

- ▶ Recertification policies
- ▶ Life cycle rules

These are described in the sections below.

1.5.1 Recertification policies

Recertification or *attestation* is a process to verify *who has access to what*. There is often a need to do recertification to comply with a company's internal security policy (for example, no unneeded accounts or accesses should be maintained active for more than three months), or you can do recertification simply because the law forces you to. The audit activity is either scheduled or *on demand*.

A *recertification policy* includes activities to ensure that users provide information that they have a valid, ongoing need for accounts or access rights.

The recertification policy defines, for example, how frequently account owners must certify their need for continued account access. Additionally, the policy defines the operation that occurs if the recipient declines or does not respond to the recertification request. Recertification policies use a set of notifications to initiate workflow activities that are involved in the recertification process. For example, a system administrator of a specific service can create a recertification policy for the service that sets a 90-day interval for account recertification. If the recipient of the recertification declines recertification, the account is automatically suspended.

Table 1-2 describes some recertification policies in more detail.

Table 1-2 Sample recertification policies

Schedule	Target	Policy
Quarterly at 03:00	Active Directory accounts	The manager of the account owner is requested to approve the recertification. If the recertification is rejected the account will be suspended. A rejection e-mail will then be sent to the account owner. If the manager has not responded within 10 days the recertification request will be automatically approved.
Monthly on the first day at 01:01 a.m.	Access to project directory on Linux file system	Request that the owners of the account recertify their continued business need for access to the project directory. If the recertification is rejected the access will be marked as rejected for recertification. A rejection e-mail will be sent to the owner of the access. If the account owner does not respond to the request the access will automatically be removed.

1.5.2 Life cycle rules

Life cycle rules provide administrators with the ability to define life cycle operations (automated processes) to be executed as the result of an event. Life cycle rules are especially useful in automating recurring administrative tasks. For example:

- ▶ Password policy compliance checking
- ▶ Notifying users to change their passwords before they expire
- ▶ Identifying life cycle changes such as accounts that are inactive for more than 30 consecutive days
- ▶ Identifying new accounts that have not been used for more than 10 days following their creation
- ▶ Identifying accounts that are candidates for deletion because they have been suspended for more than 30 days
- ▶ When a contract expires, identifying all accounts belonging to a business partner or contractor's employees and revoking their access rights

Table 1-3 describes sample life cycle rules in more detail.

Table 1-3 Sample life cycle rules

Event	Life cycle rule	Life cycle operation
Daily at 12:01 a.m.	Password expiration.	Search all account entities for the Tivoli Identity Manager and the Tivoli Access Manager services and generate an e-mail for all user accounts where the password will expire within the next seven days. Where the password is more than 45 days old, suspend the account.
Contract expires	Suspend contractor accounts.	Search for all accounts defined for a specific contractor and suspend them at the close of business on the day that the contract expires.

1.6 Access control models

This section looks at some of the access control models that are commonly found or are planned for use with a centralized identity management solution.

Note: There are many resources available that address access control models. For our discussion we are referring to the *CISSP All-in-One Exam Guide* by Harris. Another reference is the National Institute of Standards and Technologies at:

<http://www.nist.gov/>

1.6.1 The role-based access control model

Role-based access control, as its name suggests, is the granting of access privileges to a user based upon the work that they do within an organization. This allows an administrator to assign a user to single or multiple roles according to the work she is doing. Each role enables access to specific resources.

RBAC examples

Some RBAC examples are:

- A new customer** Alex registers with an organization by completing a form on a Web site. As a result of doing so, Alex may be awarded the role of *customer* by the central user administration system that in turn populates Alex's account to all customer-facing resources.
- A new employee** Betty, on starting with an organization, could be awarded the role of *basic user* by the administrator and as a result, her account information could be populated to the network access system and to an e-mail system. Betty may not yet have interacted with any of the systems, so in this case, the administrator must assign the accounts with a default password and ensure that each system makes Betty change her password upon first access.
- A senior employee** Charles would already have the basic user role from the time when he joined the organization. His work now requires that access be granted to applications that are not included within the basic user role. If he now needs access to the accounts and invoicing systems, Charles could be awarded the *accounting* role in addition to the basic user role.

A manager

Dolly would already have the basic user role from the time when she joined the organization and may also have other roles. As she has been promoted to a management post, so her needs to access other systems have increased. It may also be, however, that her needs to access some systems, as a result of her previous post, are no longer appropriate in her management role. Thus, if Dolly had basic user and accounting as her roles before promotion, it may be that she is granted the *manager* role, but has her accounting role rescinded. This would leave her with the basic user and manager roles suitable for her post.

1.6.2 Other access control models

There are two other access control models that are often found in use. These are the Discretionary Access Control (DAC) and Mandatory Access Control (MAC) models.

DAC

The DAC model is when the owner of a resource decides whether to allow a specific person access to her resource. This system is common in distributed environments that have evolved from smaller operations into larger ones. When it is well managed, it can provide adequate access control, but it is very dependant upon the resource owner understanding how to implement the security policies of the organization, and of all the models it is most like to be subject to *management by mood*. Ensuring that authorized people have access to the correct resource requires a good system for the tracking of leavers, joiners, and job changes. Tracking requests for change is often paper driven, error prone, and can be costly to maintain and audit.

MAC

The MAC model is where the resources are grouped or marked 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. User privileges to view certain resources are dependant upon that individuals clearance level.

1.6.3 Which model

All three models discussed previously have pros and cons associated with them. Which model an organization uses depends upon a number of factors, including, but not limited to, 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.

MAC

The key to this kind of system is the ability to use background security checking of personnel to a greater level than that which would normally be carried out in a business or non-governmental environment. It is also key for data of different sensitivity to be kept segregated. For example, a user must not be able to cut and paste information between documents of differing sensitivities. This has traditionally been achieved by keeping data physically separate. In this environment, therefore, a user may have a number of different workstations, one for restricted, one for secret, and so on, each running on completely different and separate architectures.

Conducting identity management across multiple sensitivity silos with one central identity management system raises a number of issues. The central system itself must be classified at the highest level, as it holds user rights to all sensitivity silos. Normally, in this environment this would mandate that various security certifications and accreditation processes have been completed and also that any cryptographic keys are in hardware storage.

As the Web portal approach matures, this kind of multiple silo approach may change, but in the short term, this would mean that a software-only solution would not be possible.

One further approach would be to treat each sensitivity silo as a discrete identity management problem. This would mean that there is a distinct solution for each silo and that the best access control model could be chosen from the other two. For example, at the lowest sensitivity silo, there are likely to be many more users that best fit an RBAC solution, while at the top level, there are fewer users and other (physical, procedural, personnel, and technical) more rigorous controls, so a DAC might be more appropriate.

Despite its limitations, this type of access control model will continue to be used in military and government environments because it provides the solid foundation for segregation of information based upon sensitivity. Identity management solutions for this space are probably best focused on the lower sensitivity silo, unless approvals can be gained to connect all silos with a highly secure management layer that includes identity management.

DAC

Discretionary Access Control is the model that is most likely to be used as a default or evolved decentralized access control solution. Organizations are familiar with the concept of each application administrator or owner being responsible for granting access to the application or system owned or administered by them. Key features of a centralized identity management system that allows this to continue are the ability to specify over-arching corporate security policies, combined with the ability to delegate responsibility for account management to individual systems. A centralized identity management system with these features allows for a reduction in the amount of management by mood, but ensures that corporate security policies can be applied, while allowing a degree of actual and real political ownership of the target resource.

Table 1-4 compares the different access control models.

Table 1-4 Access control model comparison and notes on desirable features

Access control model	Pros	Cons
MAC	<ul style="list-style-type: none">▶ Ideally suited to military and government security requirements.▶ Highly secure.	<ul style="list-style-type: none">▶ Costly to implement because of personnel vetting and data segregation requirements.▶ Difficult to centrally manage all identities because of sensitivity silos.
DAC	<ul style="list-style-type: none">▶ Likely to already be in use.▶ Easy-to-implement centralized identity management solution.▶ Suited to most commercial organizations, prior to centralized identity management or during conversion to RBAC.	<ul style="list-style-type: none">▶ Subject to management by mood.▶ Policy enforcement and audit costly.▶ Centralized identity management possible but less return on investment than single RBAC model.

Access control model	Pros	Cons
RBAC	<ul style="list-style-type: none"> ▶ Useful for strong role-focused organizations. ▶ Useful for organizations with high staff turnover and reliance on temporary or casual staff. ▶ Recommended for large user populations, particularly where users include clients and partner organizations. 	<ul style="list-style-type: none"> ▶ RBAC design can be difficult politically and logically. ▶ Strong policies required particularly where delegated administration is used.

1.6.4 Selection process

The following questions and comments are some of the thought processes that are used to help choose an access control model and centralized identity management system. Figure 1-2 on page 25 and the questions following it show and describe the path through the maze. Local, particularly non-functional requirements, may modify the approach that you must take.

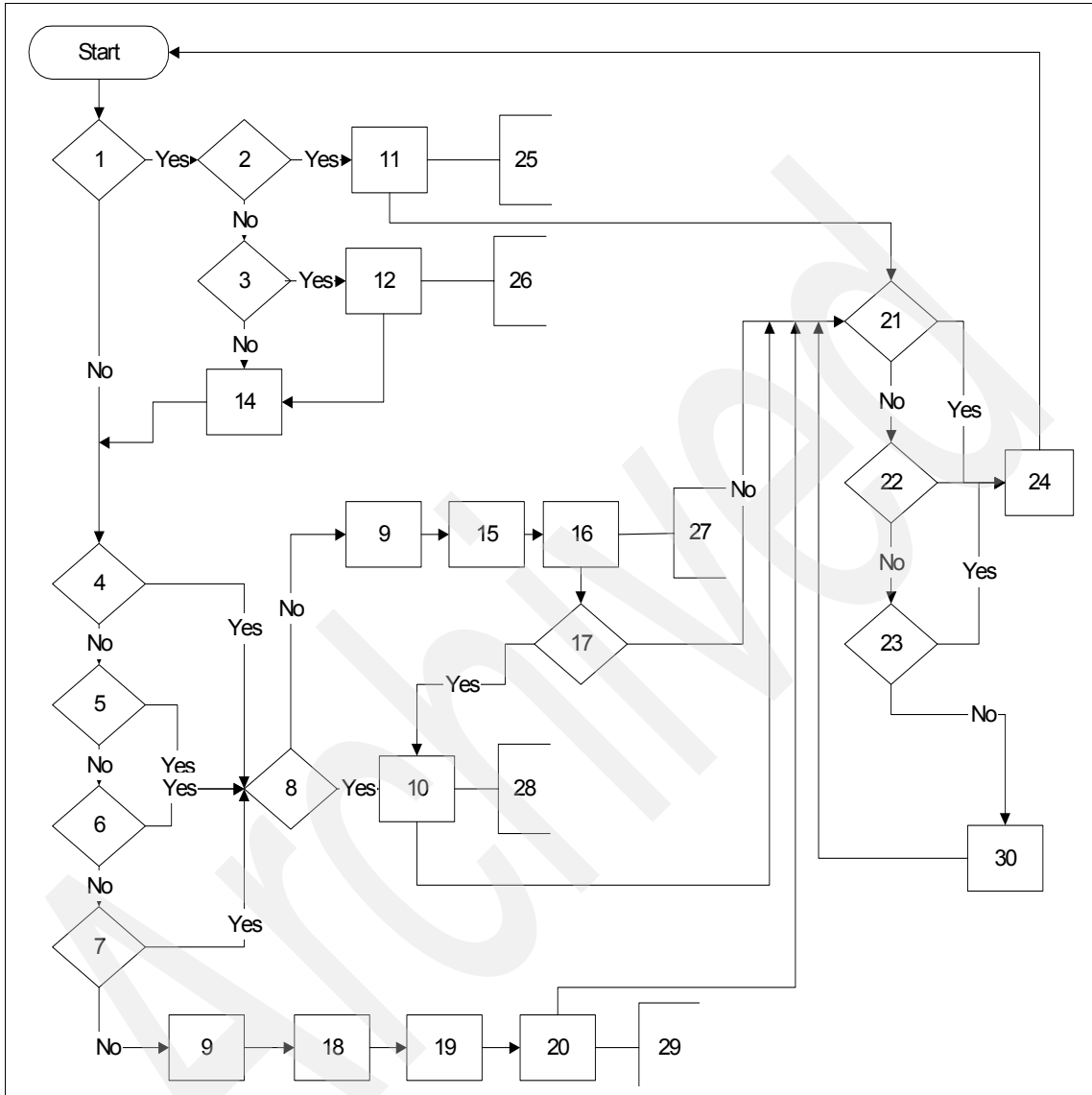


Figure 1-2 Selection flow diagram

Key questions and comments:

1. Does your organization mandate the use of sensitivity silos (confidential, secret, top secret, and so on)?

2. Your organization mandates the use of the sensitivity silos. Does it approve the use of one centralized identity management solution bridging all of the sensitivity silos?
3. If you cannot bridge the sensitivity silos with one solution, the only option is to treat each silo as a separate organization. Will your organization change its policy on the single centralized identity management system to allow bridging in the future?
4. Does your organization have a high staff turnover, or have a large number of contractors or outsourced staff?
5. Is your organization large or does it have multiple geographies that are self managing?
6. Does your organization already have a centralized or metadirectory in place or is it planning one?
7. If your organization is already using the DAC model with resource owners/administrators managing the identities of users, you could use a centralized solution to imitate this or you could move to an RBAC solution. Do you want to see further ROI and increased security?
8. If you chose to fully implement an RBAC model, will the political and business structures within your organization fully support the design work involved?
9. DAC design selected.
10. RBAC design selected.
11. Implement a single centralized identity management system with users assigned access rights based upon their approval to access one or many sensitivity silos. This is a simple form of RBAC with one role per sensitivity silo. It would be possible to make the silo model more granular, but this may detract from the essentially simple nature of the implementation. It should be noted that a user with access to one silo will gain access to all information within that silo, and therefore, in its purest form, this architecture does not address the issues of privacy or *need to know* management.
12. You can implement an identity management solution in each sensitivity silo, but should your organization's policy change, you will be able to place a master Tivoli Identity Manager over the existing silo Identity Managers to gain maximum ROI. You should therefore select a centralized management solution that is capable of supporting a hierarchy of identity management systems.
13. If you have reached this point on the flowchart, then it is probably time for a cup of coffee or a break. There is no step 13 on the flowchart.
14. Treat each sensitivity silo as a discrete problem and analyze the RBAC/DAC requirements for each silo.

15. This selection is DAC. Make sure that the centralized identity management tool that you selected has the capability to securely delegate the administration of users to the resource owner through an interface that does not require onerous training or need a thick client to be distributed. Administration of the users should be delegated to the owners of the resources. Delegated resource control should be in line with corporate policies. Centralized audit for non-compliance reports should be submitted to the resources owner regularly for her action.
16. Once deployed, assistance should be given to those business units that wish to develop an RBAC model within their *owned* space. In addition, maintaining up-to-date business cases and continuing to win greater political influence for the RBAC model should be attempted.
17. Has sufficient political ground been gained to implement an RBAC model?
18. Your organization has chosen to use DAC, which will not allow for some of the ROI traditionally associated with RBAC. Other product features also show savings, however, and you should favor products with good feature/function coverage in these areas.
19. Workflow processing. The automation of the business processes for new hires and so on should be seen as a priority. Reducing the waiting time for provisioning new users reduces productivity losses.
20. Even though DAC is the organizational model, it may still be possible to create savings by using limited or default roles. For example, every new user would automatically get LAN and e-mail accounts set up, while other systems remain within the purview of the resource owners.
21. Has a period of more than 12 months passed since you last checked the identity management system design?
22. Have any major infrastructure changes within your organization's operational systems taken place?
23. Has the nature of the external threat you face as an organization changed significantly?
24. A change has occurred within your operating environment or a long period of time has passed since you last validated your identity management system decisions. Run through the algorithm again to check on your design and amend it, if appropriate.

25. You have selected a very simple type of RBAC to map onto the MAC model in place within your organization. This means that you will also be placing increased reliance on your personnel and the vetting processes applied to them. It is possible to improve the silo granularity, but it takes time to design this granularity. Other software and hardware involved with privacy management and networking, for example, may already be in use within your organization. These should be factored into any design and planning for the solution.
26. The selection flowchart seems to suggest that you will be treating each of the sensitivity silos as a discrete identity management problem, but that you may in the future get approval to bridge the silos. The suggested method is to use a hierarchy, but if budgets and operational requirements allow, you could also scrap the existing system and replace it with a single central identity management model.
27. Reaching this point in the flowchart has meant that owing to political limitations within your organization, you have been forced to use the DAC model rather than the RBAC model, which you might naturally have selected. Using DAC, however, should be seen as a stepping stone towards RBAC. In simple terms, allowing the business owners to use the system may enable them to create roles for their own systems. It may be possible to consolidate these local roles into larger ones as time passes.
28. As you move into the real design and planning work involved in an RBAC scenario, many of the customer business units will be asked for their input about the role design problem. It may only be at this point that customer business units realize the exact impact of what you are proposing upon their *rights* to manage their own systems in their own way, regardless of the organization's security policies or of the costs involved. If this happens, you should return to question 8 and answer no to that question.
29. The DAC has been selected and the focus has been on methods (other than RBAC) of saving costs. Do not lose sight of the fact that having a central tool also brings centralized audit capabilities that will improve the security of an organization. This risk mitigation, while difficult to quantify, still improves the viability of a business.
30. Wait one month before continuing. This ensures a revalidation of your identity management strategy every month. The length of time chosen should be less than one year, but is at the discretion of your organization, taking into account all of the threat, risk, and resource issues that you face.

1.6.5 Roles versus groups

One of the difficulties that identity management system designers are facing is the way in which the terms *groups* and *roles* are used, often interchangeably or without a true understanding of their significance. They are defined as follows:

Roles A role is specifically a description of a type of user that must be provisioned to one or more services or resources.

Groups A group is specific to a target resource. It contains a subset of the users provisioned to that resource and grants access rights to a part of the resource.

Figure 1-3 shows the relationship between users, roles, services, and groups.

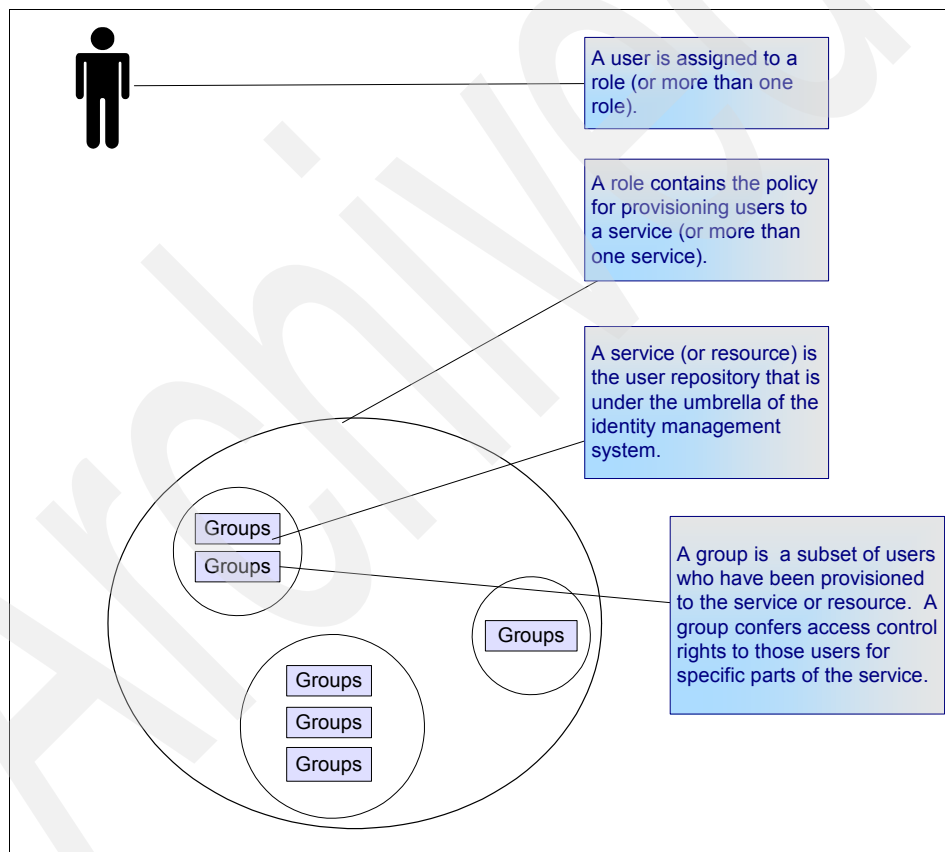


Figure 1-3 User/role/service/group relationships

Many identity management systems allow the users to be assigned to roles and hence provisioned to services. In addition, they can also provision users directly to services complete with group membership.

You can therefore use these systems to merely provision users directly to services, which is done in the absence of a valid RBAC design or in the case of the use of pure DAC.

You can also design the RBAC system such that one service is represented by one role. If each role represents only a single application, OS, database, and so on, then it is technically still an RBAC system, but it is functionally closer to a DAC system. This model is sometimes found within organizations that have not been able to successfully overcome the underlying politics. They can therefore claim to have upset no one and to have implemented a full RBAC system. The downside of this is that you have spent the time and resources on implementing an RBAC system that will not deliver the expected ROI. This model is therefore pointless and not recommended unless political considerations are more important than cost concerns.

1.6.6 Designs

The process of designing an RBAC system is fairly straightforward. If we had only two services to access (service A and service B), then users could be placed into one of three roles: Role 1 (service A only), role 2 (service B only), or role 3 (services A and B). Therefore, to access two services, there are three possible roles:

- ▶ One role containing two services
- ▶ Two roles containing one service

Similarly, to access three services, there are seven possible roles:

- ▶ One role containing three services
- ▶ Three roles containing two services
- ▶ Three roles containing one service

To access four services, there are 15 possible roles:

- ▶ One role containing four services
- ▶ Four roles containing three services
- ▶ Six roles containing two services
- ▶ Four roles containing one service

As the number of services increases, so do the potential number of roles. By the time twenty services are required (a lot less than the average number of services in a standard organization), there are 1,048,575 possible roles. It is clearly not practical to create all the possible roles and populate them. We must reduce the number of roles to those required rather than to all those possible.

It seems that a common-sense approach would be to list all the user repositories and then list all the users along with their account requirements. An example of this kind of approach is shown in Table 1-5.

Table 1-5 User to repository mapping

User	Repositories				
	NT	Internet customer application	SAP	E-mail	UNIX®
Alwena	Yes	No	Yes	Yes	No
Brian	Yes	No	No	Yes	Yes
Carmen	No	Yes	No	No	No
Daphne	No	Yes	No	No	No
Elizabeth	Yes	No	No	Yes	No
Francesca	Yes	No	No	Yes	No
Geoff	No	Yes	No	No	No
Helen	Yes	No	No	Yes	No
Ian	Yes	No	No	Yes	No
Jolina	Yes	No	Yes	Yes	No
Katya	Yes	No	No	Yes	Yes
Carmen	No	Yes	No	No	No
Mike	Yes	Yes	Yes	Yes	Yes
Neil	Yes	No	Yes	Yes	No
Ondine	No	Yes	No	No	No
Peter	No	Yes	No	No	No
Queenie	Yes	Yes	Yes	Yes	Yes
Ray	Yes	No	Yes	Yes	No
Sarah	No	Yes	No	No	No
Thomas	Yes	No	No	Yes	Yes
Uist	Yes	No	No	Yes	No
Vera	No	Yes	No	No	No

User	Repositories				
	NT	Internet customer application	SAP	E-mail	UNIX®
William	Yes	No	No	Yes	Yes
Alma	Yes	Yes	No	Yes	No
Yvette	Yes	No	No	Yes	No
Zach	Yes	No	No	Yes	Yes

Grouping these roles into similar access requirements reveals that there would be six logical roles. So in this example, five services give rise to six roles instead of all 31 possible roles, as shown in Table 1-6.

Table 1-6 User to repository mapping with roles

User	Role	Repositories				
		NT	Internet customer application	SAP	E-mail	UNIX
Elizabeth	Basic	Yes	No	No	Yes	No
Francesca	Basic	Yes	No	No	Yes	No
Helen	Basic	Yes	No	No	Yes	No
Ian	Basic	Yes	No	No	Yes	No
Uist	Basic	Yes	No	No	Yes	No
Yvette	Basic	Yes	No	No	Yes	No
Mike	CEO	Yes	Yes	Yes	Yes	Yes
Queenie	CEO	Yes	Yes	Yes	Yes	Yes
Carmen	Customer	No	Yes	No	No	No
Daphne	Customer	No	Yes	No	No	No
Geoff	Customer	No	Yes	No	No	No
Carmen	Customer	No	Yes	No	No	No
Ondine	Customer	No	Yes	No	No	No
Peter	Customer	No	Yes	No	No	No

User	Role	Repositories				
		NT	Internet customer application	SAP	E-mail	UNIX
Sarah	Customer	No	Yes	No	No	No
Vera	Customer	No	Yes	No	No	No
Alma	EMP & CUST	Yes	Yes	No	Yes	No
Alwena	HR	Yes	No	Yes	Yes	No
Jolina	HR	Yes	No	Yes	Yes	No
Neil	HR	Yes	No	Yes	Yes	No
Ray	HR	Yes	No	Yes	Yes	No
Brian	System Admin	Yes	No	No	Yes	Yes
Katya	System Admin	Yes	No	No	Yes	Yes
Thomas	System Admin	Yes	No	No	Yes	Yes
William	System Admin	Yes	No	No	Yes	Yes
Zach	System Admin	Yes	No	No	Yes	Yes

This is fine for 26 users and five services, but the next problem that emerges is one of scale. The mere collection task involved for 1,000 users or a larger range of services becomes costly and, in larger cases, unrealistic. What is needed is a single data source that is collected automatically and contains all user/service information, which can be used for reporting and analysis. Many centralized identity management solutions provide this kind of collection and reporting facility. As we have seen in an earlier section, one way of countering the political objections to RBAC is to implement centralized identity management and progress towards RBAC as political support is developed. Once again, deployment of a centralized identity management solution can be used as a tool to develop a design for an RBAC model prior to the deployment of the RBAC model itself.

There are a few other items to be careful of:

- ▶ No matter how you collect the information, it must be correct at the point of collection. Examination of the user information in Table 1-6 on page 32 would suggest that Queenie and Mike both have identical roles, in this case, CEO. In practice, however, Queenie has full access because she is the CEO, while Mike has been with the organization since leaving school and acquired a number of access permissions, as he has moved jobs within the organization and his access rights have not been rescinded. He is not the CEO.
- ▶ Similarly, Uist and Yvette both have the basic role, but neither has worked for the company for over a year. Both these cases highlight the need to carry out a reality check audit as part of the process of designing an identity management system (whether or not it is RBAC).
- ▶ Some services may have no IT dependencies. If a service is provisioned and the provisioning results in the involvement of a physical process (smart card generation and issue, uniform manufacture, and so on), then care must be taken not to include these potentially time delayed tasks into a workflow, which could delay other provisioning requirements. An RBAC design should take this type of service into account.
- ▶ Up to now, we have talked about a service as though it were one repository. We know, however, that repositories can have subsidiary groups. Most resource targets can define at least two groups (administrators and users), so in practical terms, the five services used in the 26-user example would be 10 services and have a potential 1,023 possible roles.
- ▶ Xerxes seems to be in a role of one person. He has picked up this unique role because he is both a basic employee of the organization and he is also a customer. We must therefore check with the security policy to see whether he is allowed this *double* role under one name. It makes sense in some organizations to specifically separate basic and customer roles and disallow the EMP & CUST role.
- ▶ Even if an immediate RBAC design cannot be achieved, some roles should be self-evident. A basic corporate employee user (with network and e-mail access) and an eCustomer role (with e-business application access) are examples. Implementation of these roles will serve to stimulate the RBAC design process and reduce the scale of the problem.

In practice, given the likely scale of most RBAC designs, it is necessary to include costing associated with the collection, clean up, and analysis of the existing user/repository data. We strongly recommend that any centralized identity management solution chosen be capable of being deployed as a tool to help with the design of the full RBAC model. While this RBAC design is in preparation, some ROI can be gained from the automation of user provisioning and workflow processes.

1.6.7 Observations

Most enterprises use a blend of access control models based on the sensitivity of the information or the level of effort required to change the applications. Ideally, the enterprise should have a predominant access control model such as RBAC and use the other access control model to handle exceptions. As a rule of thumb, the 80/20 ratio may be used. However, this ratio will vary based on the enterprise's business policies and security policies.

1.7 Identity management versus meta directory

Identity management and user provisioning are often lumped together with directory strategy and also meta directories. This problem arises because most people have slightly differing definitions of each of these areas and each OEM vendor selects slightly different feature/function sets to be implemented within their product or solution. This results in a lot of overlap and confusion.

Figure 1-4 on page 40 shows how some of the features of both identity management and directory strategy requirements map onto an idealized set of products.

Typically, most organizations start with many directories and either a requirement to reduce operating costs with user provisioning tools or a strategic vision for a single universal directory/repository, like x.500 or Microsoft Active Directory. These two approaches are typified by teams, such as the strategic directory team or the user provisioning project. Their titles indicate the direction that they are likely to take.

Directory strategy teams are predisposed to recommend single directories (x.500, RACF, MS Active Directory, and so on) as the solution to an organization's needs, while user provisioning teams have a tendency to recommend tools that are essentially best to address the help desk costs or user password reset problems. Some organizations even appoint teams of both types. The teams may or may not adequately communicate their plans with each other. This can lead, in the worst cases, to political control battles for the ownership of the space, or at best, an agreement not to tackle the areas common to both teams, thus leaving an unaddressed set of problems.

It is much better if organizations appoint a strategy/project team whose scope spans both user repositories (directories and meta directory strategy) and the tools needed to manage them effectively and efficiently (user provisioning and identity management). There must be a representative from the organization's security team appointed to this type of project team.

Table 1-7 and Figure 1-4 on page 40 describe some of the tool sets that should be considered. OEM products or solutions may not map exactly to this broad definition set, and organizations may not need to cover all these areas in one deployment. What is key, however, is that consideration be given to all of these areas as one integrated project and design exercise.

Table 1-7 Pros and cons of tools used in identity management and directory areas

Product/solution type	Notes	Pros	Cons
Single directory	A single repository is mandated, for example, x.500, RACF, and MS Active Directory.	All users are defined in one place, and audit, user management, and reporting are less costly. Security policies can be applied in one place.	A single directory may require the purchase of administration and management toolkits. Many applications may have to be rewritten or customized to allow authentication or authorization against the directory.
Many directories	Normally, the state of an organization itself generates the need to look strategically at the problem. The situation often has evolved rather than been designed and controlled.	High degree of flexibility. Little or no design effort required.	Costly to manage. Subject to management by mood. Difficult to audit and apply a security policy. More subject to human error. Longer provisioning time scales resulting in decreased user productivity. Less secure because of orphaned accounts.

Product/solution type	Notes	Pros	Cons
Meta directory	<p>A true meta directory is effectively a complete copy of all the user repositories within an organization held in one place. The copy is created using a set of rules contained within a join engine.</p>	<p>This tool allows for the creation of a single user directory from the one already in existence.</p> <p>It allows applications written to use a different repository to continue to do so.</p> <p>It allows business units to manage users with the existing tools, allowing the meta directory to cope with the creation of a central directory.</p>	<p>Still has multiple points of administration, so costs are not reduced.</p> <p>The central directory schema definition and creation of rule sets can be complex and therefore costly.</p> <p>Implementation time scales and future flexibility can be unacceptable.</p> <p>Can result in a large, non-performing centralized directory.</p>
Virtual meta directory	<p>The virtual meta directory is similar to a meta directory, but it does not create a complete copy of the multiple user repositories. Rather, it relies upon its join engine to perform organization-wide distributions of changes detected in the directories under its control.</p>	<p>Has all the benefits of a true meta directory without any of the same performance limitations.</p> <p>Will be able to cope with future deployment of a single directory.</p>	<p>Still requires the definition of a rule set.</p> <p>User management tools maybe limited.</p> <p>May not have real integration points with identity management tools.</p> <p>May still require a specialist to maintain and manage.</p>

Product/solution type	Notes	Pros	Cons
User administration	<p>User provisioning tools can be thought of as a tools to perform a one-way automated push of users held in a central repository (often LDAP) out to target systems. Some implementations have the ability to manually retrieve users, but this is usually limited.</p>	<p>These kinds of tools have been around longer than some of the others. The amount of user experience is therefore greater.</p> <p>Many of the other tools, by definition, have user provisioning built in. There may be no need, therefore, to consider this type of product.</p>	<p>Usually based upon a proprietary framework that may need to be deployed.</p> <p>GUIs are often thick client applications.</p> <p>Old technology that is being supplanted by more functional identify management solutions.</p> <p>Integration points to the other parts of the solution stack are often limited.</p> <p>May not be able to cope with the requirement for Internet user registration in terms of scale.</p>

Product/solution type	Notes	Pros	Cons
Identity management	<p>Can be thought of as user provisioning plus. The use of generic protocols (http, https, and so on), in addition to better integration with other products, makes this solution the most fully functional.</p>	<p>Widest range of features based upon modern non-proprietary standards.</p> <p>This allows better integration with other tools (for example, Virtual Meta Directories) and other pieces of the security architecture (for example, centralized authentication and authorization systems).</p> <p>The GUI interface is Web based, which requires a lower skill set user for all interactions.</p> <p>Self-service and delegated user management, allowing partners, users, and suppliers to self manage their information.</p> <p>Approval workflow engine to authorize data changes.</p>	<p>Many solutions (particularly those in user provisioning space) use this title for their products.</p> <p>Coverage of target platforms must be checked, particularly where no integration point with Virtual Meta Directories exists.</p>

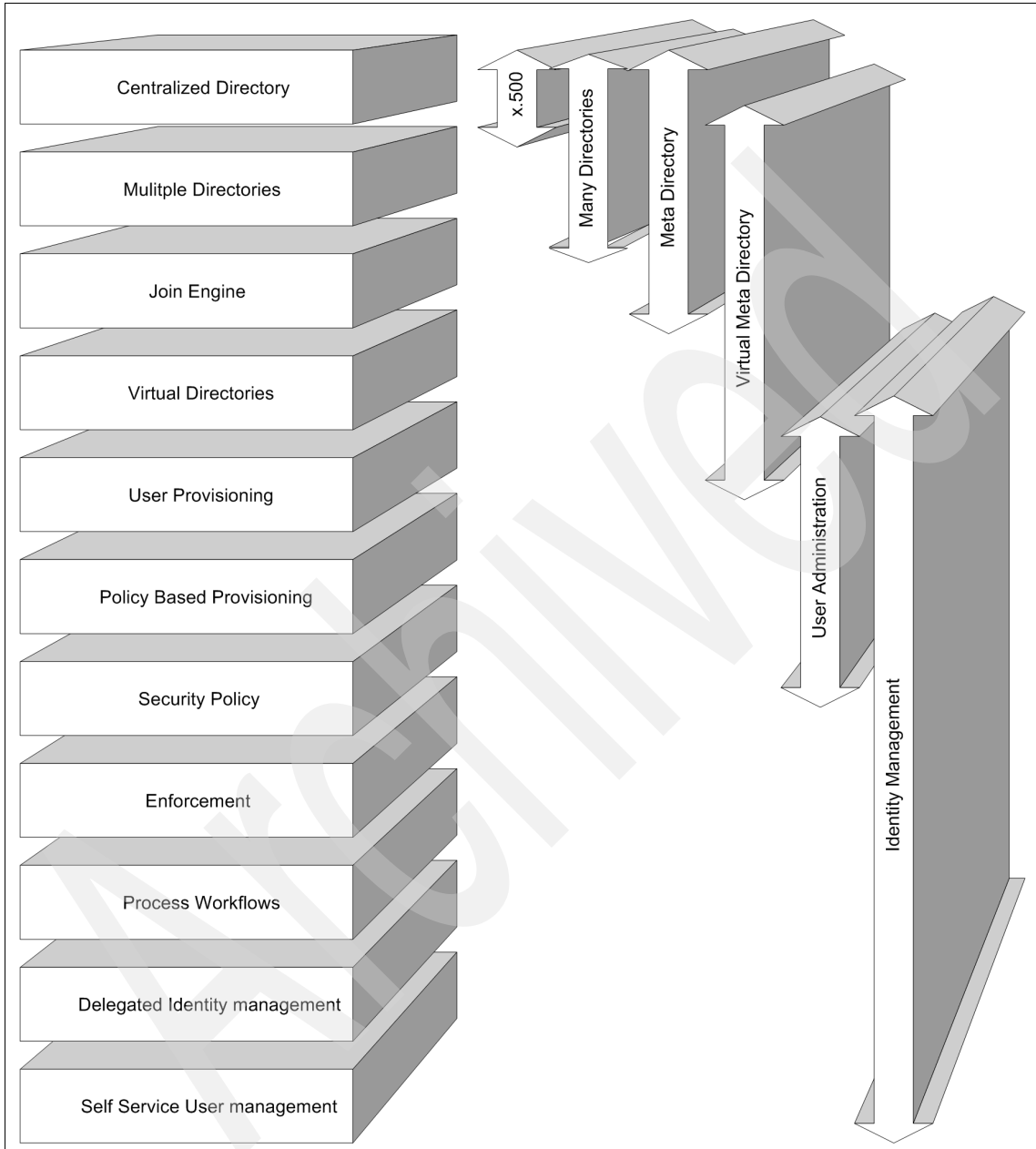


Figure 1-4 Identity management features (blocks) mapped against solution types (arrows)

“WEM Shipyards” on page 41 provides a (fictitious) example of an organization and three potential approaches to its problems to help clarify the approaches to this problem set.

WEM Shipyards

WEM Shipyards are a small but growing ship builder. In recent years, new management has bought more technology to the organization, which has resulted in increased efficiency and more orders. They are currently working on three vessels with orders for seven more. Those vessels are:

- ▶ Naval craft - HMS Nonesuch: A new mine countermeasures vessel that will also have a patrol boat role. WEM Shipyards is the lead contractor for this vessel, but much of the technology, sensors, and weapons systems are designed and fitted by sub-contractors who need access to the WEM Shipyards IT systems.
- ▶ Americas Cup yachts - Project Doris: A pair of 12 meter racing yachts designed for competition in the next Louis Vuitton Cup and the Americas cup. The customer requirements include a high level of secrecy surrounding the design, particularly the hull below the waterline.
- ▶ Traditional gaff cutter - Elizabeth Anne: A new yacht built along the traditional lines of a Bristol Pilot Cutter, but with modern electronic navigation, communications, and control systems. It will be designed for safe long-distance cruising for two or three people. If successful, WEM Shipyards hopes to market the design.

WEM Shipyards has noted that they must continue to modernize their technology and keep the cost of ownership to a minimum if they are to continue to win orders. One part of this approach is to address the cost associated with IT provisioning. In particular, user management is becoming a cost burden.

They tried using a directory strategy approach and then a user provisioning approach. Both approaches revealed weaknesses, some of which are shown below.

- ▶ **Directory strategy**

Having defined the IT requirement for a single directory, WEM Shipyards addressed a number of nonfunctional requirements and found that the team building the Elizabeth Anne had some specialized sail-making software that could not easily (and therefore without cost) be configured to use the designated single directory.

The same team also wanted to continue publishing information and progress reports on the build on the Web, but had plans to create an application to ask users to register for access. This would be used to help market future builds of this class. The potential for the number of registered users from this Internet-facing application was thought to be large and management of these users must be an IT function, as the build team did not have the skills or resources. But IT was not comfortable allowing Internet users direct access to their central directory for authentication/authorization.

The team working on project Doris said that their customer was uncomfortable with the use of a single-user repository, particularly when some of the subcontractors working on HMS Nonesuch were also working on other yachts competing for the next Americas Cup. They required that the specialized hull design software and therefore its data be treated separately from the single directory approach.

The sub contractors assisting in the build of HMS Nonesuch were happy with the idea of a single directory, as it appeared to give them a more efficient working interface with WEM Shipyards, while the management overheads would be entirely borne by WEM Shipyards.

WEM Shipyards realized that they could not implement a single central directory. At best, they could manage a central directory with three others:

- Sail design application repository
- Internet LDAP directory
- Hull design repository for Doris

They concluded that in any single directory implementation there would always be external requirements that mandated multiple directories and therefore, at best, an 80/20% solution.

► User provisioning

This approach found that although there were some improvements over the existing system of user management, there were a number of things that stalled the project.

The costs associated with managing sub-contractors defined within the system still remained with WEM Shipyards. More importantly, the customer for HMS Nonesuch was concerned that there was no workflow/approval process to register a user on the system, and all that was required was a request from a sub-contractor. Under the old system, a manual process had been in place, but because of the automation, it was easier for a user to be provisioned and bypass that process.

User provisioning was one-way only. This meant that the administrators of the target platforms, who might still change user details locally, could corrupt the system integrity, as there was no detection/synchronization method within the solution set.

► Holistic approach

Shipyards came to the conclusion that this issue was too large to address all at once. As a result, they have chosen to address each issue one at a time. By selecting two or possibly three solution set products and implementing them over time, they are able to gain maximum commercial advantage. The steps to addressing these issues are:

- a. Implement an identify management solution (which shows the greatest ROI), as this allows them to control their overhead.
- b. Integrate the system with a virtual meta directory as a second project, which would allow WEB Shipyards to extend the scope of the solution.
- c. Consolidate towards a single user repository. This step is not seen as being needed internally, but it is likely that this project would start as a result of external pressures, such as a customer requirement, general changes to the global IT environment, or, more realistically, as a result of a successful merger or takeover bid.

Reaching this conclusion was only possible because they took the holistic approach and critically examined all the options. The selection of products must be done on the basis that full integration, while not an immediate requirement, is definitely mandated.

1.8 Identity management provisioning models

Depending on business needs, identity management provides the alternatives to provision resources to authorized users on a request-based, role-based, or hybrid approach.

Figure 1-5 illustrates the different provisioning models.

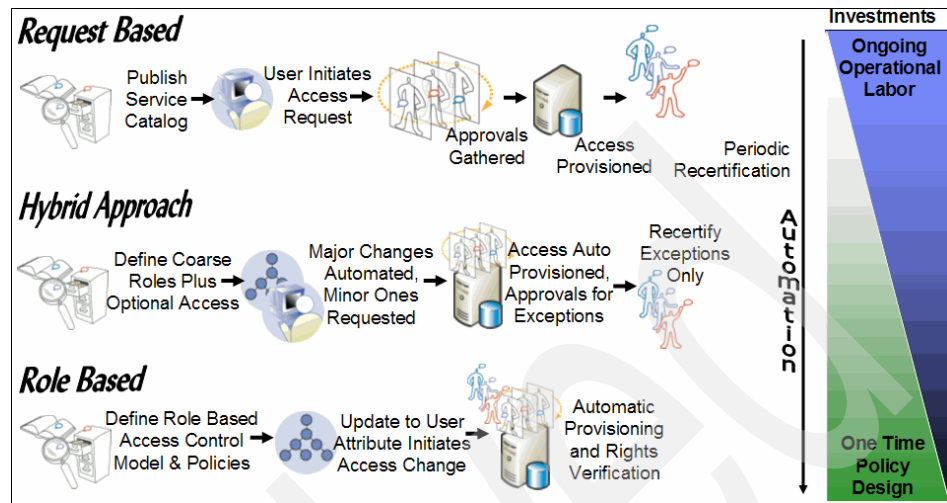


Figure 1-5 Identity management provisioning models

1.8.1 Request-based approach

On a request basis, an identity management system should be able to provide a process to grant, modify, and remove access to resources throughout a business and to establish an effective audit trail using automated reports.

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. For example, users or their managers can request access to new accounts. Additionally, managers or other administrators are alerted to unused accounts and given the option to delete the accounts through a recertification process. These periodic reviews of user access rights ensure that previously approved access is removed if it is no longer needed.

1.8.2 Role-based approach

Using *role-based provisioning*, businesses can automate and accelerate the process of granting access to resources and lower the risk of individuals gaining more system access than required by their job or other relationship to a company.

The operational needs of an enterprise will determine the assignment of users to organizational roles. For example, a user might have a role as a help desk

assistant or auditor. In a role-based model, users receive a specific set of accounts and access rights based on role membership. When a user is removed from a role, the entire set of accounts and access rights are also removed.

1.8.3 Hybrid approach

The *hybrid* model of provisioning resources combines the request and role-based approaches.

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

Other companies might find it impractical for business reasons to achieve complete role-based provisioning, and so target a hybrid approach as a desired goal. Still other companies might be satisfied with only request-based provisioning, and not wish to invest additional effort to defining and managing role-based, automated provisioning policies.

Whatever approach a business decides to use, the identity management system should support the use of it.

1.9 Role management

Depending on the access control model adopted by the organization, it is crucial to properly manage roles and their relationship within the model if the organization chooses to adopt the role-based approach.

1.9.1 Relationship and hierarchy

Depending on the complexity of the organization, there can be several relationships between roles. For example, if there is an engineering department, users in this department should have access to several applications. Some users in this department have a design role (access to design applications) and others have a mechanical role (mechanical applications), but both roles could have access to the same set of base engineering applications; in this example, design engineer and mechanical engineer roles are thus inherited privileges from the more generic engineer role. The inheritance defines the basis of the hierarchical relationship in which both mechanical engineer and design engineer are children of the engineer role.

In general, in role-based access control, roles can be business related or application related. Business roles are mapped to application roles in order to grant access and usage privileges. This situation can become quite complex when it comes time to manage users and roles, because the number of each one tends to grow once you start looking into more fine grained details.

When it comes to the provisioning model, an organization depends on the selected access control role model, because these models are based on policies, and policies closely rely on roles. As shown in Figure 1-5 on page 44, there are three approaches to selecting an identity management provisioning model.

- ▶ The request-based approach establishes that an identity management system should be able to provide a way to define processes that grant, modify, and remove access to resources throughout a business and to establish an effective audit trail using automated reports. This means that in some cases, and at specific times, a user could request access for every application and role within the organization.

Because all forms of access are opened for users to request, there are some considerations that have to be taken into account before taking this approach. For example, does a user actually have access to this specific system? Why do they need additional access? If they already have access to the system, but they need an extra role on this system, how is the actual access related to the requested one?

At this point, it is important that the identity management system be able to manage relationships in a role hierarchy.

- ▶ Following the role-based approach, an organization can automate and accelerate the process of granting access to resources and lower the risk of individuals gaining more system access than required by their job or other relationships within the organization.
- ▶ The hybrid approach combines the request and role-based approaches. As we discuss in 1.8.3, “Hybrid approach” on page 45, an organization might want to automate access provisioning for a subset of employees or managed systems by using role-based assignments, and also handle all other access requests or exceptions through a request-based model. At this point, it is crucial how the organization tree and the role hierarchy are designed.

1.10 Separation of duties

As R. A. Botha and J. H. P. Eloff say in the IBM Journal Of Research and Development:

“Separation of duty, as a security principle, has as its primary objective the prevention of *fraud* and *errors*. This objective is achieved by disseminating the tasks and associated privileges for a specific business process among multiple users. This principle is demonstrated in the traditional example of separation of duty found in the requirement of two signatures on a check. Previous work on separation of duty requirements often explored implementations based on role-based access control principles. These implementations are concerned with constraining the associations between RBAC components, namely *users*, *roles*, and *permissions*. Enforcement of the separation of duty requirements, although an integrity requirement, thus relies on an access control service that is sensitive to the separation of duty requirements.”⁸

As we mention in 1.9, “Role management” on page 45, the complexity within the role hierarchy will vary from one organization to another. Besides the role hierarchy and the relationships between roles, it is necessary to establish controls for roles that conflict with each other.

For example, if you are an employee of a company who needs to claim business travel expenses, you need access to the claim application as a normal user. There has to be another role defined in this process that approves the expenses, and that role must not be the same, even if both users are using the same application. Separation of duty policies are there to ensure that the same employee cannot be assigned to both roles simultaneously. Many business processes, however, require that there be a possibility of allowing temporary exceptions to these rules, for example, in the case of sickness, extended travel, or similar reasons. These exceptions will have to be properly recorded and documented to adhere to regulatory controls or policies.

1.11 Conclusion

Let us finally take a look at general goals of using a centralized identity and credential management infrastructure:

- ▶ Easing compliance with security audits
- ▶ Standard and custom reporting
- ▶ Consolidating control of the user management processes

⁸ Find more information at <http://www.research.ibm.com/journal/sj/403/botha.html>.

- ▶ Eliminating inconsistencies from human error and *management by mood*
- ▶ Reducing training costs and education requirements
- ▶ Reducing help desk and overall administration costs
- ▶ Involving fewer people in day-to-day management and redeploying them to higher value assignments
- ▶ Dividing work along organizational/departmental structures
- ▶ Improving response to user changes
- ▶ Leveraging user information in all business processes

Table 1-8 summarizes the business benefits of an identity and credential management solution.

Table 1-8 Tivoli Identity Manager benefits

Features	Advantages	Benefits
Centralized Web (HTML) administration interfaces	Provides ubiquitous management interfaces and centralizes the definition of users and provisioning of user services	Reduces the education and training costs and complexity associated with managing from multiple native interfaces, while leveraging the consolidated repository of user information.
Role-based delegated administration	Enables delegation of administrative privileges along organizational and geographical boundaries	Accommodates political or distributed management needs.
Self-service interfaces	Enables users to perform password resets, password synchronization, and modifications to personal information without administrative intervention	Helps reduce help-desk costs and eases the burden of daily administration on help desk and IT staff.
Embedded workflow engine	Automates the submission and approval processes for access requests and changes to user information	Helps decrease the potential errors and inconsistency common to manual business processes.
Life cycle rules	Automates administrative tasks based on an event or time	Consistent policy enforcement plus the timely removal of incorrect entitlements and inactive accounts.
Standard reporting	Monitors entities as near to the time of updating as possible	Demonstrates compliance with security policies while providing an audit trail.

Features	Advantages	Benefits
Embedded provisioning engine and application management toolkit	Automates the implementation of administrative requests on the environment, and provides a mechanism for extending the management model to support new and custom environments	Helps increase potential productivity and reduce administrative overhead, while supporting new business initiatives as the company grows.
Delegated administration	Allows an organization to offload some of the day-to-day workload and therefore costs	Changes made to accounts by delegated administrators still must conform to the security policies in force.
Multiple repository support	Includes all user repositories in the coverage of identity management solutions, thus reducing the cost of specialist administrators	Including all user repositories in the coverage of identity management solutions allows policy to be applied uniformly.
Centralized auditing and reporting	Reduces time spent on following audit trails on disparate systems	Centralized auditing makes the tracing of events more realistic and therefore much more secure.

Archived

Architecting identity and credential management solutions

In order to design and architect general enterprise security solutions, you should utilize proven methodologies and concepts. In this chapter we discuss the approach for architecting an identity and credential management system as being part of an overall enterprise security architecture as well as the aspects of understanding and re-engineering enterprise business processes for managing identities and credentials.

The background information about how to architect an enterprise security architecture using IBM Method for Architecting Secure Solutions (MASS) can be found in a special edition of the IBM Systems Journal on *End-to-End Security*, Vol. 40, No. 3¹ and in IBM Redbooks publication *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014. Details on business process management can be obtained from IBM Redbooks publication *Continuous Business Process Management with HOLOSOFX BPM Suite and IBM MQSeries Workflow*, SG24-6590.

¹ Copyright 2001 International Business Machines Corporation.

2.1 Solution architectures, design, and methodologies

This IBM Redbooks publication focuses on the design of an identity and credential management solution. In this section we look at how to go about producing an identity management solution design.

Building the design is just one part of the implementation of a solution. So we look at the design as part of the implementation. Following that, we discuss the identity management solution design and methodologies.

2.1.1 Overview and terminology

There is much confusion surrounding the terminology for architecture and design. The terms *architecture* and *design* are often used interchangeably. Within the generic term *architecture* there is an endless list of terms used, which may include:

- ▶ Functional architecture
- ▶ Operational architecture
- ▶ Physical architecture
- ▶ Logical architecture
- ▶ Enterprise IT architecture
- ▶ Security architecture
- ▶ Messaging architecture
- ▶ Database architecture
- ▶ Web hosting architecture
- ▶ Systems architecture

While there are no standard, universally accepted definitions, there are many good terms and many of them mean roughly the same thing.

In *Software Architecture in Practice, Second Edition*, by Bass, et al, software architecture is defined as:

“The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.”

Thus, an architecture can be thought of as the bits and how they fit together.

Architectures will often be high-level, such as enterprise-wide, and not concerned with product details. We are concerned with producing a document detailing how a product-centric identity management solution will be implemented: Thus we are concerned with both architecture and design:

- Architecture** Describing how the bits of the product fit together, and any integration with external components, at an abstract level and at a physical level.
- Design** How each of the bits, and any integration, are configured or customized to meet the existing system environment and solution requirements.

Architectures often present a number of views of the structures in a system. A product-centric architecture, such as an identity management architecture, is normally concerned with two architectural structures: The conceptual (or logical) structure and the physical structure. Bass, et al., give the following definitions for these:

- Conceptual, or logical, structure** The units are abstractions of the system's functional requirements. These abstractions are related by the shares-data-with relation. This view is useful for understanding the interactions between entities in the problem space and their variations.
- Physical structure** This view shows the mapping of software onto hardware. It is particularly of interest in distributed or parallel systems. The components are hardware entities (processors) and the links are communication pathways. Relations between the processors are communicates-with. This view allows an engineer to reason about performance, availability, and security.

There are a number of other structures described by Bass, et al., which may be relevant to a product-centric architecture, but we are only concerned with the logical and physical structures.

The following terms are used throughout this book. We attempt to be consistent (although this is not guaranteed to be consistent with other IBM/Tivoli books).

- Solution** This is the product-centric identity management system that we are designing.

Architecture	An architecture describes the structure of the system and the relationships between components.
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.
Design	The configuration or customization of specific components or sub-components within the solution to adhere to an existing system environment or solution requirements.

Be aware that different methodologies use different terminology. When discussing the methodologies in the following section, other terms are used to be consistent with the methodology.

2.1.2 Implementation flow

Any implementation will be part of a project and follow a standard set of steps or phases. Most projects are subject to methodologies defining the phase execution and deliverables. There may be a number of methodologies involved, such as a project management methodology (for example, the IBM WorldWide Project Management Methodology (WWPMM)) and one or more design and implementation methodologies (such as IBM Method for Architecting Secure Solutions (MASS)). There may be deployment methodologies that are product specific for the software selected that leverage the supplier's intellectual capital such as best practices, lessons learned, and tools for estimating hardware or automating parts of the installation process.

The implementation may use multiple methodologies depending on the project's scope and complexity in addition to the skills of the people responsible for the implementation. The most successful implementations that deliver the most value to the business utilize a multiple-discipline team that leverages the unique perspectives and skills of each person. A sample multiple-discipline project team may consist of:

- ▶ Executive project sponsors from the affected business units and the IT department
- ▶ Business unit owners
- ▶ Project manager
- ▶ IT architect
- ▶ Solution designers for each component of the architecture
- ▶ Deployment subject matter experts for each component of the architecture
- ▶ IT security administrator
- ▶ IT operations personnel
- ▶ IT auditor

In this book we are concerned with the product-centric architecture and design for an identity management product. The project to produce the product-centric architecture normally follows one or more of the following processes:

- ▶ An enterprise conducts an enterprise-wide software architecture project to review the entire IT environment and produce an enterprise-wide IT architecture. The resulting architecture may dictate the need for a solution around identity management.
- ▶ An enterprise conducts an enterprise-wide security architecture project. The project looks at all security aspects of the enterprise (not just the IT security). The resulting architecture identifies the security areas where the enterprise must focus. This may include identifying the need for a solution around identity management. This exercise often also produces the enterprise security policy document that dictates the security policy to be applied to an enterprise, its employees, and its customers.
- ▶ An enterprise purchases an identity management solution based on specific business needs, such as cost cutting, audit compliance, legal compliance, or consistent application of corporate standards.

These lead to a project to deploy an identity management solution, which includes developing a product-centric architecture and design document. This is shown in Figure 2-1.

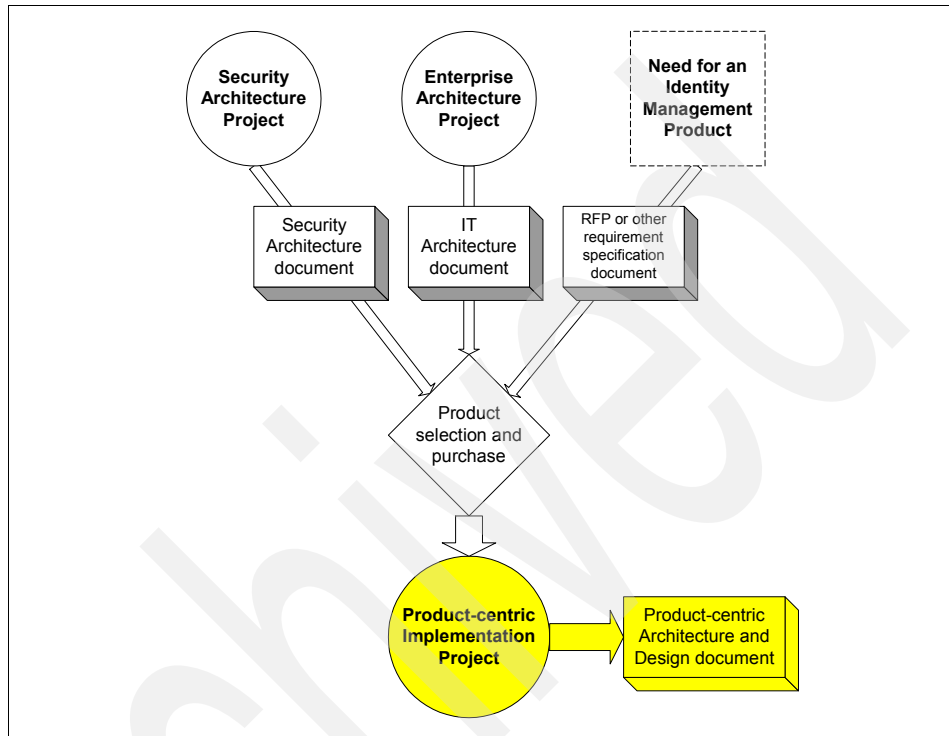


Figure 2-1 High-level and product-specific projects

Most projects involve business tasks (such as cost-benefit analysis and budgeting), project management tasks (such as scheduling, resource allocation, and risk management), and technical tasks (such as design, build, test, and deploy). We restrict our discussion to the technical tasks associated with the production of the architecture and design document. Figure 2-2 shows a set of generic steps or phases that relate to the architecture and design document.

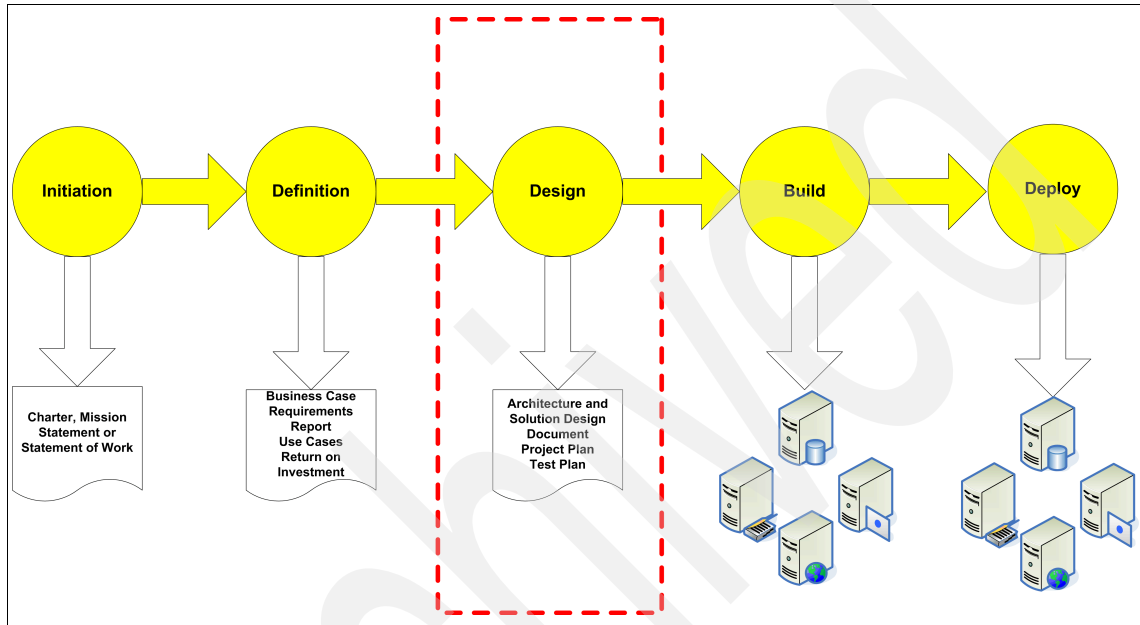


Figure 2-2 Generic implementation phases for a project

The steps are:

Initiation

This is the project initiation step. It normally involves identifying the project background and requirements at a high level. The deliverable for this step is some sort of statement of work (SoW) or mission statement. The high-level requirements will have come from a preceding project (such as an IT architecture or security architecture project) or the software purchase requirements.

Definition	This is the project definition step, for example, where the project is defined in detail. This involves gathering the details (the existing systems, users, procedures, and other information and the detailed requirements of the solution). The deliverable for this step will be one or more documents defining the project. These may include a project definition report, the business case, a requirements document, a use case, a return on investment (ROI) statement, a functional specification, and an existing system analysis document.
Design	This is the design step. It involves designing the solution. The deliverable for this phase is the architecture and design document. Consideration must be given to the success criteria and how to verify that is has been met through comprehensive testing procedures.
Build	This is where the solution is built and tested prior to going into production.
Deploy	This is where the solution is deployed in production and where steady state operations are maintained.

The focus of this IBM Redbooks publication is on the design phase and the creation of the architecture and design document (as highlighted in Figure 2-2 on page 57). However, much of the information required for the design will have been gathered and documented in the definition phase. The next section discusses this.

2.1.3 Definition of an identity management solution

The definition phase defines the project in detail and involves detailing the current environment, the problem to be solved by the solution, and the detailed requirements for the solution.

The initial project definition will be based on the documentation that triggered this project, such as the IT architecture, security architecture, request for proposal (RFP), or equivalent. These documents identify the business background, the business need for the solution, and, usually, the business and technical requirements for the solution.

For an identity management solution, the following areas must be defined in this phase (in no particular order):

- ▶ **User management procedures:** The procedures for managing users, who manages users, and what is required of the solution for managing users.

- ▶ Password management procedures: The procedures for managing account passwords, who manages passwords, and what is required of the solution for managing passwords.
- ▶ Access control management procedures: The procedures for managing access control, who manages access control definition, and what is required of the solution for managing access control.
- ▶ Security policy: What the corporate security policy defines for users, accounts, passwords, and access control.
- ▶ Target systems: The current system environment (including operating systems, databases, applications, the network, firewalls, physical location, and access control) and the system requirements of the solution.
- ▶ Interfaces: The interfaces to the current identity management mechanisms and procedures and the integration requirements of the solution.
- ▶ Auditing and reporting procedures: The procedures for auditing and reporting, who is involved in the auditing and reporting of users and their access, the audit requirements for the solution, and the reporting requirements for the solution.
- ▶ Role management procedures: Procedures used to manage user access to resources. Role management does not grant or remove user access itself; instead, it sets up a role hierarchy to help you do it more efficiently.
- ▶ Group management procedures: Procedures used to create, change, and delete groups on target systems.
- ▶ Technical requirements: The other technical requirements for the solution, such as maintaining steady state operations like monitoring the availability of critical resources plus developing both server and data backup, restore, and recovery procedures.

Gathering this information usually involves a series of interviews and workshops with the people and teams involved in identity management. This may include the chief information officer (CIO), IT executive, security management/administration team, IT auditor, operations, help desk, key technical teams (NT admin, UNIX sysadmin, and so on), any application development teams, and business managers involved in the project. The combination of these interviews and workshops will develop a picture of how the system currently works and how it could 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 and set the priorities, although others may 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 between the project owner and the project team prior to the commencement of the design phase. In some enterprises, there may be a formal sign-off procedure by the business executives before funds are allocated to proceed to the design phase. This formal sign-off procedure may also be required for the build and the deploy phases.

2.1.4 Design of an identity management solution

Eberhardt Rechtin, in *Systems Architecting: Creating and Building Complex Systems*, suggests an approach for developing an architecture, differentiating between the system (what is built), the model (a description of the system to be built), the system architecture (the structure of the system), and the overall architecture (an inclusive set consisting of the system architecture, its function, the environment within which it will live, and the process used to build and operate it).

For our identity management product-centric solution, the overall architecture is the product-centric architecture, for example, the architecture and design document for our identity management solution. However, as identity management is a key player in the security space, the following discussion also applies to the high-level security architecture.

Rechtin outlines the steps for creating a model as follows:

1. Aggregating closely related functions
2. Partitioning or reducing the model into its parts
3. Fitting or integrating components and subsystems together into a functioning system

The security system model will be represented by the aggregation of security functions, expressed in terms of subsystems and how the subsystems interact. The security-related functions within a networked information system (NIS) can be described as a coordinated set of processes that are distributed throughout the computing environment. The notion of distributed security systems, coordinated by design and deployment, meets the intuitive expectation that security within an NIS should be considered pervasive. In an NIS environment, security subsystems must be considered as abstract constructs in order to follow Rechtin's definition.

2.1.5 Data model considerations

There are other models that must coexist with the security systems model. There is the underlying data model that holds the information for establishing a person's electronic identity and generating that person's credentials. This is called an authoritative data source. One of the first design tasks is to identify the authoritative data source, analyze its contents, and understand how and when the data is updated.

The designer must be able to answer these questions:

- ▶ Is there sufficient information to uniquely identify each person in the organization?
- ▶ Is there sufficient information to identify each person's role or job function in order to develop policies to provision accounts with the appropriate access levels to the necessary systems resources?

In many organizations there are silos of information containing the attributes that identify a person and how he contributes to the business. This information may be in a human resources management system (HRMS); it may be in assorted operational systems, such as sales forecasting, order entry, manufacturing, finance; and so on. A data mapping exercise followed by a data clean-up effort may be required to construct the authoritative data source, as illustrated in Figure 2-3.

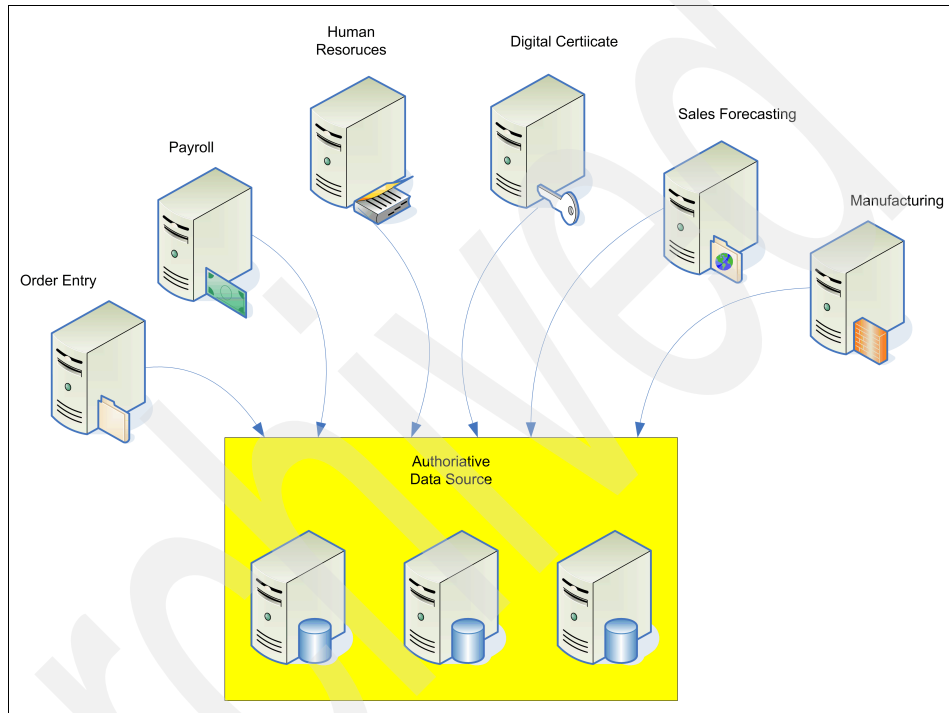


Figure 2-3 Operational systems feeding the authoritative data source

In some organizations, the human resources management system may be the authoritative data source and it may be used as a one-way data feed to the operational systems. In this case, an identity management solution may be inserted between the authoritative data source and the soon-to-be-*managed* operational systems. This is shown in Figure 2-4.

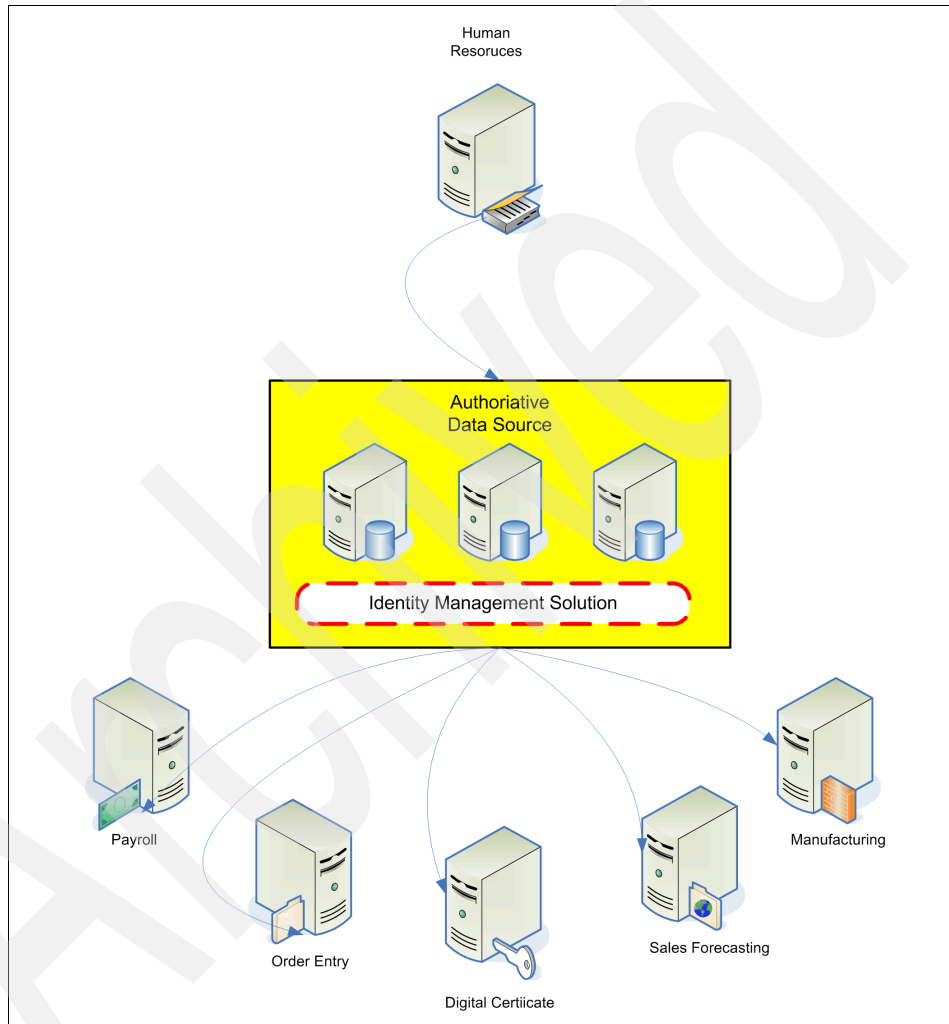


Figure 2-4 Authoritative data source feeding the IM solution and the operational systems

Understanding the data model is a key requirement for developing a successful identity management solution. The solution designer must understand the data relationships and the data update procedures in order to assess the validity of the authoritative data source. This may be a time-intensive process, so it may be advantageous to start the data mapping exercise while the functional requirements are being defined.

In most cases the data mapping exercise will identify data clean-up and data synchronization tasks that are required to keep the attribute values synchronized between the authoritative source and the operational systems. This may identify requirements to install a metadirectory tool or a virtual directory product as part of the identity management solution.

Another factor to consider is the industry trend for server consolidation. Although it is technically possible to share a directory (Lightweight Directory Access Protocol, or LDAP) or share a relational database (RDMS) between the security services and application services, it is not advisable.

The security services should be separated from the application services because the security services require a fast response and they cannot afford potential delays from sharing system resources with other running applications or large queries. Another consideration is that the application may be the most vulnerable component to unauthorized access, and this may adversely affect the availability of the server and compromise the confidentiality of the data.

2.1.6 Migration considerations

Most enterprise deployments have two instances of the identity management solution:

- ▶ A non-production, test, and development instance
- ▶ A production instance

Depending on the enterprise's requirements for testing before production activation and their resources, they may have additional instances of the identity management solution:

- ▶ A non-production system's integration test instance
- ▶ A non-production quality assurance test instance
- ▶ A non-production staging or pre-production instance that mirrors production

In Table 2-1 we describe the purposes of the different instances.

Table 2-1 Identity management instances

Instance	Purpose
Development	A small-scale prototype, using a representative subset of the production data where new functions including software updates are introduced and unit tested.
Systems integration testing	A small-scale model of production, using a representative subset of production data with interfaces to servers that provide data feeds to the identity management solution and servers whose accounts are managed by the identity management system.
Quality assurance testing	A controlled and small-scale model of production similar in scope and size to the systems integration testing instance (above). However, this instance may be used for training administrators and users, plus developing documentation.
Pre-production	A locked-down and controlled mirror image of the production environment where new applications are thoroughly analyzed to determine their impact on the production environment from the security, functional, and performance perspectives.
Production	Live <i>bet the business</i> instance.

The use of multiple instances creates an interesting dilemma. How do you manage these configurations? How do you migrate identity management objects between these systems instances without having to create them in each instance?

An identity management solution should have built-in capabilities or tools in order to export objects from one instance and import them to another instance.

2.2 Identity management design process

User and access control administration have been around for a few years. Tivoli has had the user administration and security management products for several years. This means that there is already intellectual capital in the form of white papers, best practices, and lessons learned relating to the architecture and design of the products. This could easily be extended, with the information in the following chapters of this book, to produce an Tivoli Identity Manager architecture and design. The other approach is to start from the ground up with an established methodology.

The IBM Redbooks publication *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014, introduces the Method for Architecting Secure Solutions (MASS) as a methodology for developing a design for a security implementation.

Thus, to prepare an identity management architecture and design document, we can do one of the following:

- ▶ Rely on past experience, existing intellectual capital, and a generic design approach where we map the customer environment and requirements to product functionality.
- ▶ Use a methodology such as MASS.
- ▶ Combine the approaches.

However, MASS is also very strong in preparing the high-level security architecture. This is shown in Figure 2-5.

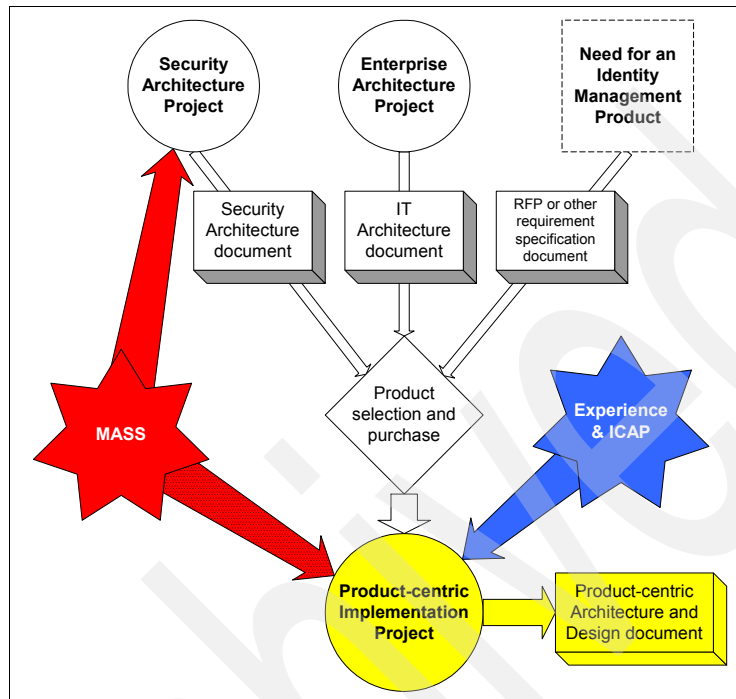


Figure 2-5 Methodologies and projects

MASS is particularly strong on the high-level security architecture and where there are multiple security functions covered by a solution. However, the MASS components do not map exactly to the Tivoli security products. In particular, the MASS identity and credential management subsystem does not map well to a pure Tivoli Identity Manager deployment, but may map to an integrated Tivoli Identity Manager and Tivoli Access Manager deployment.

The following sections are extracts from the IBM Redbooks publication *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014, that apply to an identity management solution design.

2.2.1 MASS and identity management

In this section, we summarize MASS and how it maps to identity management.

MASS subsystems

For this project, Common Criteria were considered to be the description of the complete function of the security system model. The classes and families within the Common Criteria represent an aggregation of requirements; however, after careful review, it was determined that the class and family structures defined within Common Criteria do not lend themselves to be used as part of a taxonomy for pervasive security. The aggregation is more reflective of abstract security themes, such as cryptographic operations and data protection, rather than security in the context of IT operational function. To suit the objective of this project, the Common Criteria functional criteria were re-examined and reaggregated, removing the class and family structures. An analysis of the 130 component-level requirements in relation to their function within an NIS solution suggests a partitioning into five operational categories:

- ▶ Audit
- ▶ Access control
- ▶ Flow control
- ▶ Identity and credentials
- ▶ Solution integrity

A summary mapping of CC classes to functional categories is provided in Table 2-2.

Table 2-2 Placing Common Criteria classes in functional categories

Functional category	Common Criteria functional class
Audit	Audit, component protection, and resource utilization
Access control	Data protection, component protection, security management, component access, cryptographic support, identification and authentication, communication, and trusted path/channel
Flow control	Communication, cryptographic support, data protection, component protection, trusted path/channel, and privacy
Identity/credentials	Cryptographic support, data protection, component protection, identification and authentication, component access, security management, and trusted path/channel
Solution integrity	Cryptographic support, data protection, component protection, resource utilization, and security management

While redundancy is apparent at the class level, there is only a small overlap at the family level of the hierarchy defined within Common Criteria and below. Much of the overlap represents the intersection of function and interdependency among the categories.

The component-level guidance of Common Criteria documents rules, decision criteria, functions, actions, and mechanisms. This structure supports the assertion that the five categories described in Table 2-2 on page 68 represent a set of interrelated processes, or subsystems, for security. The notion of a security subsystem has been proposed previously; the authors of *Trust in Cyberspace* described functions within operating system access control components as belonging to a decision subsystem or an enforcement subsystem.

The five interrelated security subsystems proposed here and depicted in Figure 2-6 expand the operating system-based concept and suggest that function and interdependency of security-related functions, beyond centralized access control, can be modeled as well.

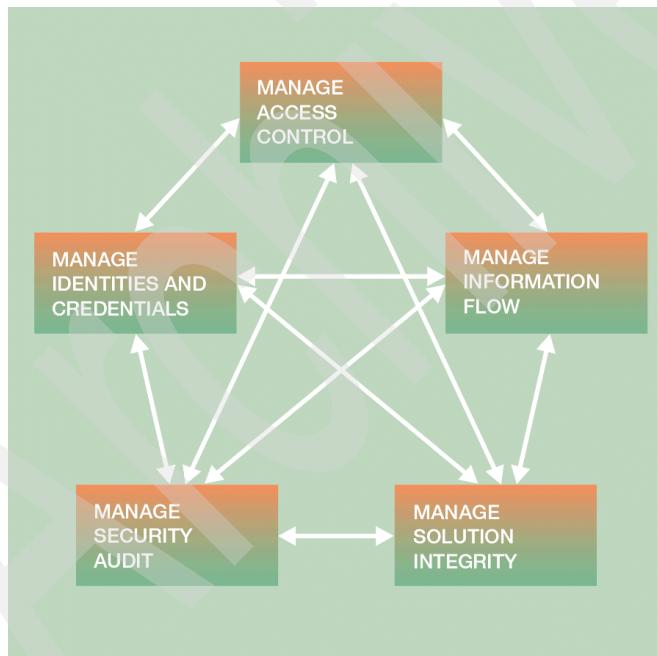


Figure 2-6 MASS subsystems

In this book, we are primarily concerned with the Manage Identities and Credentials section of MASS. This is detailed in the following section.

MASS identity and credential management

The purpose of a credential subsystem in an IT solution is to generate, distribute, and manage the data objects that convey identity and permissions across networks and among the platforms, the processes, and the security subsystems within a computing solution. In some applications, credential systems may be required to adhere to legal criteria for creation and maintenance of trusted identity used within legally binding transactions.

A credential subsystem may rely on other subsystems in order to manage the distribution, integrity, and accuracy of credentials. A credential subsystem has, potentially, a more direct link to operational business activities than the other security subsystems, owing to the fact that enrollment and user support are integral parts of the control processes it contains. From Common Criteria, a credential subsystem may include the following functional requirements:

- ▶ Single-use versus multiple-use mechanisms, either cryptographic or non-cryptographic
- ▶ Generation and verification of secrets
- ▶ Identities and credentials to be used to protect security flows or business process flows
- ▶ Identities and credentials to be used in protection of assets: integrity or non-observability
- ▶ Identities and credentials to be used in access control: identification, authentication, and access control for the purpose of user-subject binding
- ▶ Credentials to be used for purposes of identity in legally binding transactions
- ▶ Timing and duration of identification and authentication
- ▶ Life cycle of credentials
- ▶ Anonymity and pseudonymity mechanisms

The closed loop process for a credential subsystem is represented in Figure 2-7.

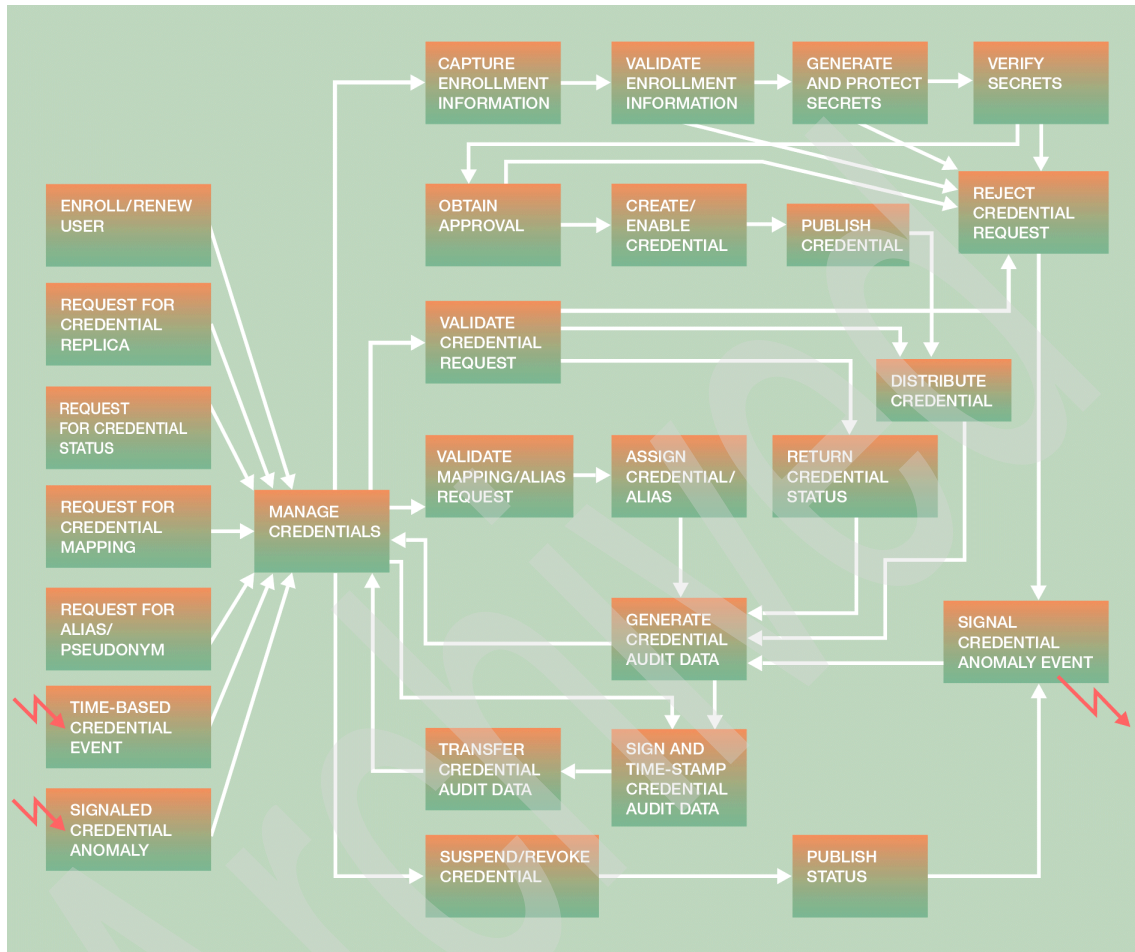


Figure 2-7 Identity and credential management subsystem

As you will see as you go through the Tivoli Identity Manager architecture and design sections of this book, not all of the components of the MASS identity and credential management subsystem are covered by Tivoli Identity Manager. When using MASS, you must be aware of what the method proscribes and what the products provide. In some cases, you can ignore the discrepancy as it may not apply. When the method proscribes a function that the implementation requires, you may have to add additional products or develop custom functionality.

2.2.2 Developing security architectures using MASS

Chapter 2, “Method for Architecting Secure Solutions,” of the IBM Redbooks publication *Enterprise Security Architecture using IBM Tivoli Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014, provides a detailed example of developing a security architecture following the MASS methodology.

The steps defined by MASS for the security architecture development are:

1. Model business processes.
2. Establish security design objectives.
3. Select and enumerate subsystems.
4. Document conceptual security architecture.

These are discussed in the following sections.

Model business processes

In this step, you model the business processes to which the security architecture must be applied. Note that these processes may not be the business processes of the identity management solution (as discussed in 2.3, “Business processes and identity management” on page 80).

Establish security design objectives

Define the design objectives for the security architecture, considering the business processes along with existing security policies and security procedures. For example, the design objectives for an identity management solution may include:

- ▶ There is a need for passwords to be managed by both users and administrators.
- ▶ There is a need for roles to be managed centrally and cover all applications, systems, and databases each role requires. Additionally, the definition of parent-child relationships within a role hierarchy has to be considered.
- ▶ There is a need for all account logins for a user to be consistent across all applications, systems, and database.
- ▶ There are security policies for password creation and setting the password change interval.
- ▶ There is a separation of duty (SoD) for roles feature, which helps prevent invalid or inconsistent combinations of roles, such as preventing parent-child relationships within an SoD policy.

These design objectives will flow from the detailed requirements.

Select and enumerate subsystems

In this step, the design objectives are mapped to the MASS subsystems. For an identity management solution, most design objectives will map to the Identity and Credentials Management subsystem. For an extended solution (such as Tivoli Identity Manager and Tivoli Access Manager), many of the subsystems may be involved. The mapping includes identifying where a subsystem is required or is supplementary.

There may need to be multiple instances of each subsystem within a design. For example, there may need to be independent identity management subsystems for external (Internet) customers and employees. Or there may need to be separate identity management subsystems for separately managed businesses. Actual subsystem enumeration requires documented rationale.

Document conceptual security architecture

With the design objectives agreed and mapped to the subsystems, a conceptual model for security within the IT solution can be created. The example in the IBM Redbooks publication *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014, shows the conceptual model as a series of diagrams, with each showing the system environment for a particular design objective and the security subsystem components mapped to it. Other representations are possible.

From the perspective of the enterprise deploying the solution, the security design objectives will dictate where security functionality is desired; however, the compliance to some or all of the security requirements may be limited by the enforceability of policies beyond the boundaries of the enterprise. Whether and how these credential subsystems and access control subsystems can be integrated into the security architecture can have a major impact on the trustworthiness of the solution as a whole. These issues and dependencies should be considered and documented within architectural decisions.

This type of conceptual model forms the baseline for developing and evaluating a proof-of-concept and further refinement of the functional aspects of security within the target environment.

Summary

The design flow and work/items or deliverables for these steps are shown in Figure 2-8.

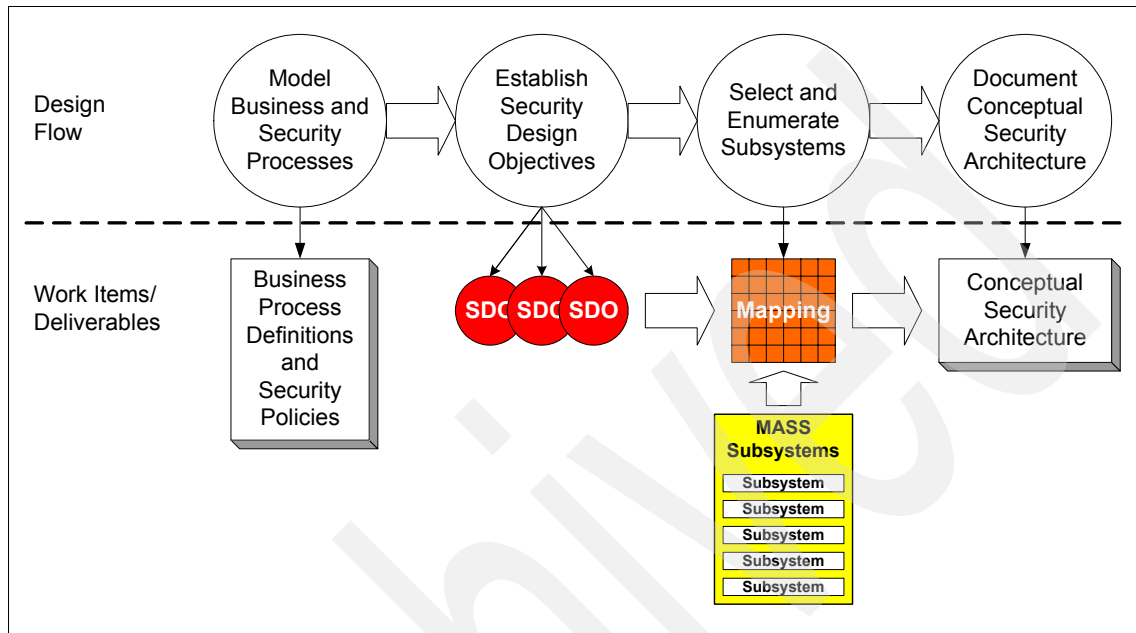


Figure 2-8 Security architecture development using MASS

Once the conceptual model has been developed, it must be integrated into the overall solution architecture.

2.2.3 Integration into the overall solution architecture

There are several steps involved in translating the conceptual security subsystem functions into component-level specifications and integration guidance. These include creating models of the solution environment, documenting architectural decisions, developing use cases, refining the functional design, and integrating security requirements into component architectures.

The following sections are extracts from the IBM Redbooks publication *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014.

Solution models

Creating an initial solution model is a critical step in the design process. With skill and experience, one-of-a-kind solution models can be developed to fit a given set of requirements. For complex solutions, the practice of using templates derived from prior solutions is becoming commonplace.

The Enterprise Solutions Structure (ESS) provides a range of reference architectures² for e-business solutions.

Documenting architectural decisions

Previously, the notion of the duality of security design was described, that is, ensuring correct and reliable operation and protecting against error and maliciousness. Both motivations are based upon managing the business risks of the solution and of the environment. Risks represent the likelihood that an undesirable outcome will be realized from a malicious attack, unexpected event, operational error, and so on. Risks are either accepted as a cost of operation, transferred to some other party, covered by liability insurance, or mitigated by the security architecture.

Architectural decisions will dictate how robust the security system architecture should be, which security subsystems to incorporate into the system architecture, which functions and mechanisms within each subsystem should be deployed, where the mechanisms will be deployed, and how the deployment will be managed.

Examples of architectural decisions include:

- ▶ Viability of the countermeasures, including the threats addressed, the limitations and caveats of the solution, and the resulting window of risk
- ▶ Extensibility of the design, including whether or not the design will serve the total population and if there will be separate designs for defined population segments
- ▶ Usability of the design, including whether or not the mechanisms integrate with the technology base and the extent of the burden of compliance for users
- ▶ Manageability of the design, including the extent of the burden of life cycle management

Use cases

Architectural decisions will also drive the evaluation of prototypes and models of functions within the solution. One form of prototype is called a use case. Both security threats and normal interactions and flows can be validated with use cases.

² P. T. L. Lloyd and G. M. Galambos, "Technical Reference Architectures," IBM Systems Journal 38, No. 1, 51–75 (1999).

There are many architectural decisions to be evaluated within each iteration of the design. The effect on performance due to processing delays, plus the effect of data collection and analysis on the overall operation of the solution, are significant factors.

Refining the functional design

Walkthroughs of complete business processes along with any associated security policies and security processes, including exception conditions and handling processes, assist in creating a viable solution outline and refining requirements and interdependencies among the solution building blocks.

Integrating requirements into component architectures

The security functions within the design must be apportioned throughout the solution. However, many of the mechanisms and services within the IT solution that implement security functionality operate within other than security components, for example, database systems, application systems, clients, servers, and operating systems.

The task of adopting security functions into the network, application, middleware, security, systems management, and infrastructure architectures is shared by the several architects and integration specialists involved in the design project. The process involves a structured approach, considering the purposeful allocation of functions and requirements throughout the component architectures by:

- ▶ Mandate, based upon a legal or contractual compliance requirement
- ▶ Compliance with enterprise security policies and processes
- ▶ Best practice for security, or for balance of security and business process
- ▶ Component capability, knowing the existence of a mechanism that supports the required process or action
- ▶ Location in the configuration, based upon interaction with components or flows
- ▶ Impact, considering the risk, security objective, or the component capacity to perform
- ▶ Necessity, because there may be no better alternative

Summary of the design process

This section has described the process for translating the conceptualized security solution into a set of detailed specifications, for an integrated IT security control system, using the security subsystem construct. The design is documented, refined, and validated against the business processes, existing security policies, and existing security processes, through use cases and scenarios. The detailed security requirements, expressed in terms of Common Criteria component-level detail, are distributed throughout the operational model for the IT solution. At this point, integration-level detail can be finalized, and the implementation plan can proceed.

Archived

The entire solution architecture process is shown in Figure 2-9 on page 79. It includes the security subsystem architecture development diagram (Figure 2-8 on page 74).

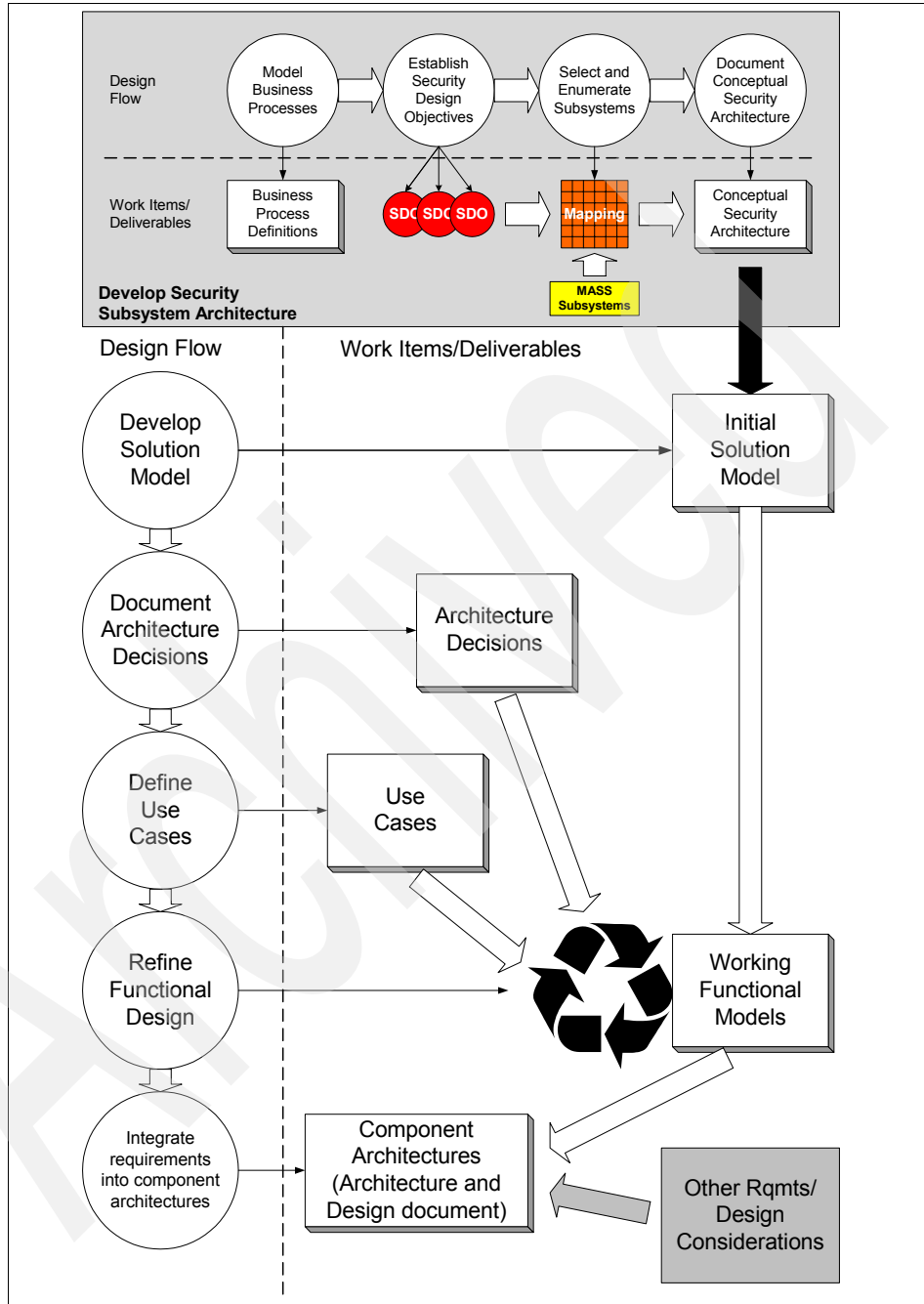


Figure 2-9 Developing an overall solution architecture

The security architecture design and the overall architecture design sections both discussed business processes and how they are integrated into the design. The next section discusses business processes in an identity management solution.

2.3 Business processes and identity management

The identity management solution will comprise both business (or procedural) and technical (security subsystem-specific) functionality. An implementation will not just involve installing an identity management tool; there will be integration with existing business procedures and their associated security policies if they exist. There is the potential for some business process re-engineering (BPR). Both technical (product-related) and business (process-related) skills will be required in the definition and design phases.

To produce an effective identity management solution, the architect must understand all identity processes involved in detail. Let us look at an example.

A new employee starts working for a company. How does their identity information get created? Is there an HR database involved? How is that connected to their salary and benefits? How does HR tie in with the IT department? How does that person get access to the applications they need to do their job?

The list of processes can include:

- ▶ A person joining a company and being defined to the HR system
- ▶ A person getting accounts to access applications
- ▶ A person getting passwords to use the accounts
- ▶ A person changing departments with bulk account changes
- ▶ A person changing a role with subtle account changes
- ▶ A person changing a surname and impacting accounts
- ▶ A person changing passwords
- ▶ A person resigning and being *marched out*, requiring locking of accounts
- ▶ A person resigning but their account must be accessed by others
- ▶ A password being reset by an administrator
- ▶ A locked account being unlocked
- ▶ An account being locked
- ▶ All accounts for a user being deleted
- ▶ A set of accounts being moved from one system to another
- ▶ An access control group being changed and impacting a number of users

This list is not exhaustive, but indicates the business process review exercise that should be performed as part of the Project Definition phase.

Implementing an identity management solution may involve designing a solution that complements the existing business and security processes or it may involve significant business process re-engineering and drive updates to the data model. The project requirements will indicate the level of business process re-engineering.

Adoption of any re-engineered processes must involve analysis of the impact of the solution on:

- ▶ The system owners. For an identity management solution, this will be the company executive (for example, the owners of the security policy) and the IT Security department.
- ▶ The system administrators. For an identity management solution, this will be the security administrators, help desk staff, and technical support.
- ▶ The system users. For an identity management solution, this will be everyone defined as IT users in an organization.

Any changes to processes could potentially impact every person in a company. These changes may drive the implementation of an identity management system (for example reducing password-reset help desk calls by allowing users to change their own passwords). If there are to be changes to the processes, the architect and project team must be cognizant of:

- ▶ Usability: Users of various skill levels may be using the solution, so the usability of the components must be appropriate to all levels of users.
- ▶ Documentation: Process changes impacting a large number of users will require greater documentation support than a change impacting a small team. This may include procedure documents, intranet pages, and online help.
- ▶ Education: As with documentation, if you are deploying significant changes to a large number of people, thought must be given to the education plan.

We are not going to discuss the business process re-engineering methodologies in this IBM Redbooks publication. There are many books and Web sites that contain such information. The following IBM Redbooks publications may be of interest:

- ▶ *Intra-Enterprise Business Process Management*, SG24-6173
- ▶ *Business Process Reengineering and Beyond*, SG24-2590

2.4 Conclusions

This chapter has examined the issues and circumstances that affect the design of an identity management solution. It has outlined a system model and a systematic process for security design with the Common Criteria international standard at its foundation.

A key component of the identity management design is the understanding, and possible re-engineering, of the business processes and the data flows associated with identity management.

The remainder of this book will present the architecture and design considerations for Tivoli Identity Manager and will build on the concepts presented in this chapter.

Tivoli Identity Manager component structure

In this chapter we introduce the high-level components and new concepts for the design of an identity management solution.

We provide you with an understanding of the following topics:

- ▶ The high-level logical component architecture for Tivoli Identity Manager
- ▶ The various internal modules and sub-processes of Tivoli Identity Manager
- ▶ The high-level physical architecture of Tivoli Identity Manager

3.1 Logical component architecture

In order to understand the IBM Tivoli Identity Manager system architecture and how to utilize its capabilities, it helps to understand the architecture logically. In the following sections we explain the logical components of the Tivoli Identity Manager architecture.

Tivoli Identity Manager can be thought of logically as having two primary areas of functionality, *presentation* and *provisioning*, where provisioning is represented by the application and service layer below. The logical component design of Tivoli Identity Manager is depicted in the center of Figure 3-1 on page 85. The layers are:

- ▶ The user interface layer
- ▶ The application layer
- ▶ The service layer

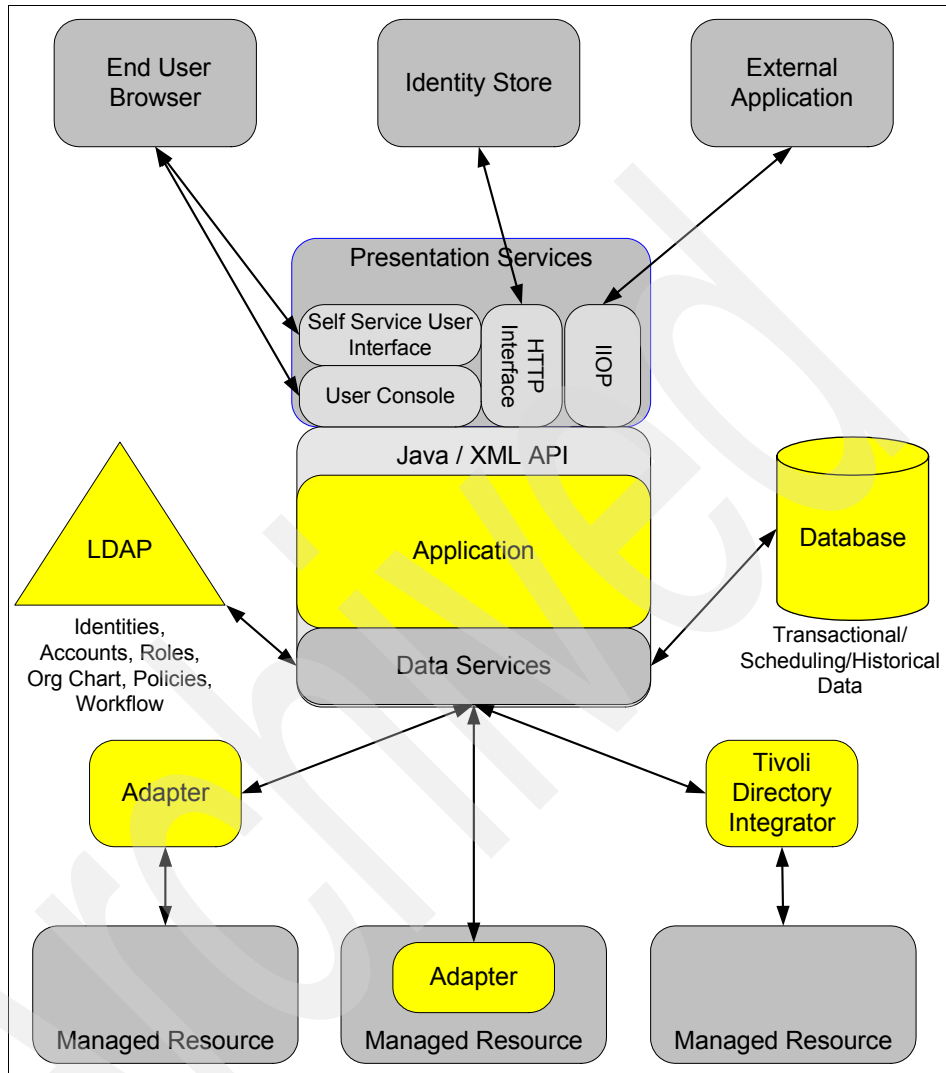


Figure 3-1 Tivoli Identity Manager logical architecture

The following sections cover each individual layer in more detail.

3.1.1 Application layer

The application layer is the interface used to access all publicly available provisioning functions.

The application subsystem contains modules that provide provisioning-specific capabilities, such as, but not limited to, identity management, account management, and policy management. Each application makes use of the core services in the services layer to achieve its goals. It is the application layer that provides the external interface to the provisioning platform. For a complete list of provisioning functions that are available publicly, refer to the Tivoli Identity Manager 5.1 Java™ Documentation, which can be found at:

<ITIM_HOME>/extensions/5.1/api/index.html

Figure 3-2 shows an overview of the different module subprocesses in the application layer. We provide more details in the following sections.

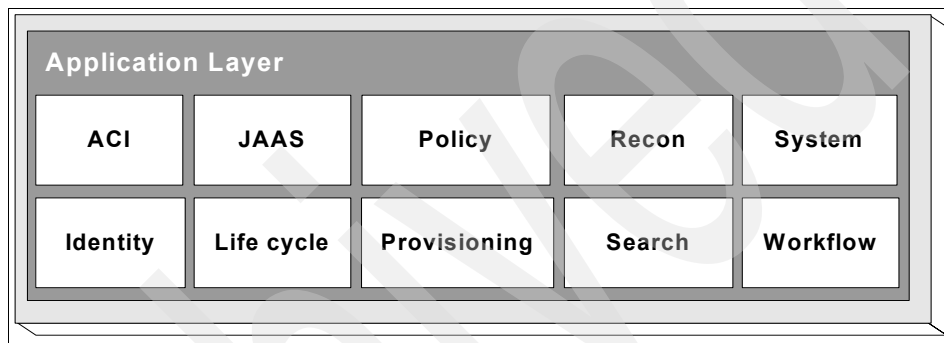


Figure 3-2 Application layer module subprocesses

Access control item (ACI)

ACI provides an interface to manage the system's access control list on a container-by-container basis, for example, basic add, list, modify, and delete operations for a remote client.

Identity management

The identity management module provides the capabilities required to manage identities, such as their addition, removal, suspension, reinstatement, transferal, and modification, including the changing of roles. The definition of roles, including dynamic roles, is also included in this module.

Java Authentication and Authorization Service (JAAS)

The JAAS module contains implementations of JAAS callback and CallbackHandler interfaces.

As in previous Tivoli Identity Manager releases, Tivoli Identity Manager's authentication module utilizes a propriety user registry in Lightweight Directory Access Protocol (LDAP), and the Java Authentication and Authorization Service

for authentication. The default JAAS implementation checks the user-supplied password against this user registry. Configuration of this JAAS implementation, in addition to configuration of some proprietary Tivoli Identity Manager settings, allows authentication for single sign-on by using Tivoli Access Manager for e-business (WebSEAL) as well.

Life cycle management

The life cycle management module provides life cycle management capabilities on supported business domain objects, including person and account. It provides an interface to invoke global operations, static operations defined for specific types of objects, or operations on the object instance. Refer to “Designing and Implementing Workflows” in the Tivoli Identity Manager InfoCenter for more information.

Policy management

The policy management module provides the capabilities to manage the policies in the system, including provisioning, password, service selection, and identity policies.

Provisioning management

The provisioning management module provides the capabilities required to manage accounts, such as their addition, removal, suspension, reinstatement, and modification.

Reconciliation management

The reconciliation management module provides service reconciliation management capabilities, as well as the capability to represent a reconciliation unit configured for a given resource or service. A resource or service can have multiple reconciliation units configured with different schedule information. So no two reconciliation units for the same resource can have the same schedule information.

Search management

The search management module provides a definable query and sort facility for defining the parameters of the search, such as a flexible string-based filter, sorting details, and the scope of the search. Furthermore, it provides interfaces for retrieving the results as a whole or by pages. There are also interfaces for resorting the results. To reduce overhead of repetitive calls back to the platform to obtain information about each object returned from a search, the results are made up of value objects directly instead of managed objects. It is still easy to obtain managed objects if needed by simply extracting the distinguished name from the value object.

System management

The system management module provides the capabilities required to manage the Tivoli Identity Manager system users and roles, such as defining behavioral properties.

Workflow management

The workflow management module provides the capabilities required to manage workflow processes, such as their addition, modification, and removal. This module also provides the ability to view the status and details of active and historical processes.

Note: Sitting between the user interface and the applications layer in Figure 3-1 on page 85 is the public Java application programming interface (API). This API provides a set of Java classes that abstract the more commonly used functions of the provisioning platform, such as identity management, password management, and provisioning management. Some of the classes that make up this API are the same classes that the Tivoli Identity Manager product uses for its base user interface.

For more information refer to documentation provided with the applications API in the <TIM_HOME>/extensions/5.1/doc/applications directory.

3.1.2 Service layer

If the Tivoli Identity Manager server is the application of complex rules that have been developed, then the applications server is the engine that runs those rules or objects. The applications server is communicating not only with the user-facing Web server, but also with the adapters residing on the managed services and with directories for storage of information.

The core services subsystem contains all modules that provide general services that can be used within the context of provisioning, such as authentication, authorization, workflow, and policy enforcement. These services often make use of other services to achieve their goals. Figure 3-3 is a graphical representation of the services interface followed by a description of each of the modules shown.

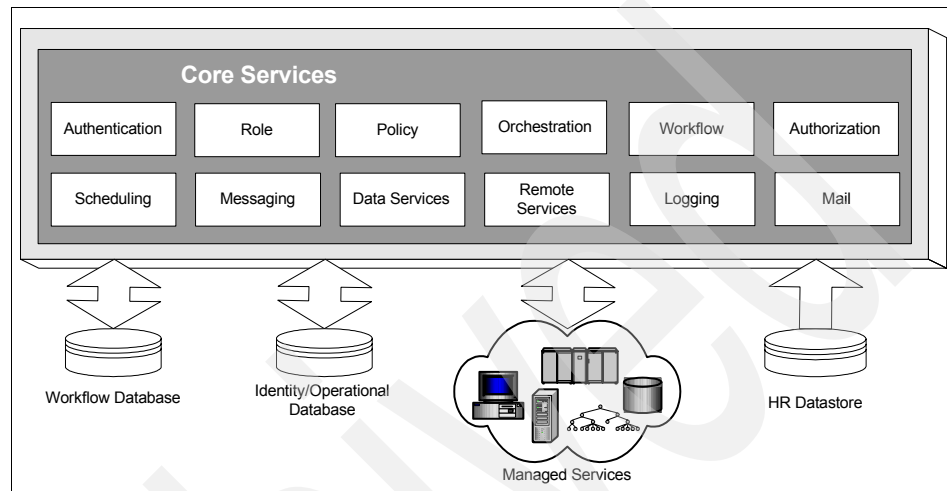


Figure 3-3 Core services module subprocesses

Authentication

The authentication module provides a set of authentication implementations that can be used by clients of the service. Examples of these implementations are simple password authentication and X.509 certificate authentication. The module is designed as a framework that can be extended by customers to provide their own implementations.

Role

The role module evaluates dynamic memberships to roles. This module is called upon when an identity or dynamic role definition changes to identify which identities should be members of dynamic roles. It is also called when security administrators build and plan logical role hierarchies or role relationships.

Policy

The policy module enforces the policies that associate users with services. This module ensures that provisioning requests conform to the policies that are defined. It resolves the appropriate policies that apply to a user and determines the services for which that user is authorized. It validates and generates passwords, and generates identities for users and accounts.

Orchestration

The orchestration module provides a coordination service for extensible operations that are performed on entities and manages the life cycles of those entities. For instance, the orchestration module provides an abstraction layer to the account management application for executing the steps needed to provision an account of a given type. Regardless of the steps involved, which could be customized or changed, the account management module would always use the same interface to the orchestration module.

Workflow

The workflow module executes and tracks transactions within the system. This includes the provisioning and deprovisioning of a service, a user's status change, the custom process associated with a provisioning request in the system, or any other transaction that affects a user's, or a group of users', access to services. Each of these transactions is persistent for fault-tolerant execution and historical auditing purposes. Clients can query the workflow module for the status of the transactions being executed.

Authorization

The authorization module provides an interface to enforce authorization rules as clients attempt operations in the system. These rules apply to accessing data within the system, as well as to operations that can be applied to the system data.

Scheduling

The scheduling module provides a timer that notifies *clients* of timed events for which they are subscribed. The scheduling module uses the messaging module to notify those clients.

Note: The *clients* discussed in this section are internal to Tivoli Identity Manager. For example, workflow is a client to scheduling. It uses scheduling to allow workflows to start at a later date instead of immediately.

Messaging

The messaging module provides guaranteed asynchronous messaging between internal modules in the architecture. This module relies heavily on the Java Message Service (JMS) specification to provide support for multiple messaging middleware vendor implementations.

Data services

The data services module provides a logical view of the data in persistent storage (LDAPv3 directory) in a manner that is independent of the type of data source that holds the data. The model abstracts the details of the stored data into more usable constructs, such as users, groups, and services. The model also provides an extendable interface to allow for customized attributes that correspond to these constructs. Metadata information about the persistent data can also be retrieved using this module.

Remote services

The remote services module provides the interaction with the external systems for provisioning and deprovisioning services. The synchronization of service information and user information is also performed within this module. The module is designed as a framework that can be extended by customers to provide their own implementations of provisioning and deprovisioning of services. This allows the platform to easily support different protocols and APIs that may be supported by the resources to be provisioned.

Logging

The logging module provides a common logging interface to all other modules.

Mail

The mail module provides an interface for notifying users via a messaging system, such as e-mail. The module is configurable to accommodate different messaging systems.

3.1.3 LDAP Directory

The Tivoli Identity Manager system uses an LDAPv3 directory server as its primary repository for storing the current state of the enterprise that it is managing. This state information includes the identities, accounts, roles, organization chart, policies, and workflow designs.

More details on the LDAP Directory and its schema are available in the *IBM Tivoli Identity Manager Database and Directory Schema Reference Version 5.1*, SC27-2413.

3.1.4 Database

A relational database is used to store all transactional, reporting, and schedule information, as well as some configuration information. Typically, this information is temporary for the currently executing transactions, but there is also historical information that is stored indefinitely to provide an audit trail of all transactions that the system has executed.

More details on the database and its schema are available in *IBM Tivoli Identity Manager Database and Directory Schema Reference Version 5.1, SC27-2413*.

3.1.5 Resource connectivity

The back-end resources that are being provisioned by Tivoli Identity Manager are generally very diverse in their capabilities and interfaces. The Tivoli Identity Manager system itself provides an extensible framework for adapting to these differences in order to communicate directly with the resource. For a more distributed computing alternative, a built-in capability to communicate with a remote adapter is provided. The adapters typically use an XML-based protocol, Directory Access Markup Language (DAML) or Remote Method Invocation (RMI) as a communications mechanism.

Remote Method Invocation (RMI) connectivity

RMI is a Java technology that allows methods of Java objects to be called from a remote location. IBM Directory Integrator uses RMI for Tivoli Identity Manager to connect to managed resources.

RMI with Directory Integrator uses a 3-layered model (stubbs layer, remote reference layer, and transport layer) that enables operations to be performed on managed resources. See Figure 3-4.

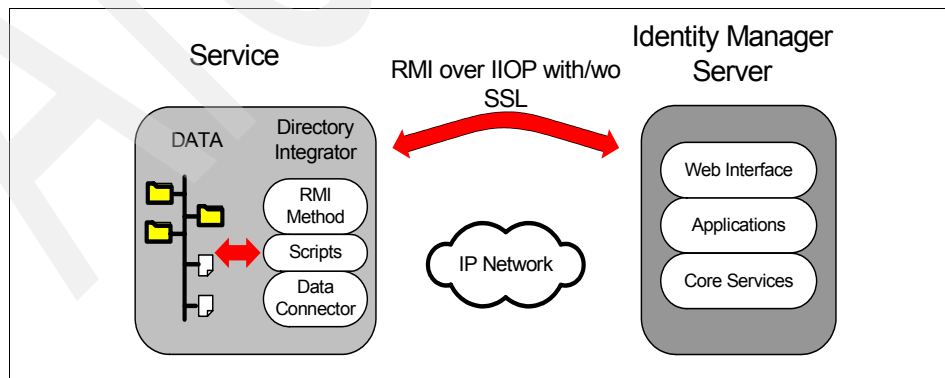


Figure 3-4 RMI communication using IBM Tivoli Directory Integrator

Tivoli Directory Integrator is a stand-alone framework for data integration. It is used to connect to host adapters. Communication to Tivoli Directory Integrator will be performed via RMI. Tivoli Directory Integrator ships with Java libraries and properties forms that allow it to be a client-initiating communication with Tivoli Identity Manager and also listen for events from Tivoli Identity Manager using RMI.

Directory Access Markup Language connectivity

DAML is a proprietary XML message format used when communicating with one of Tivoli Identity Manager's standalone adapters. These adapters are programs installed on either the managed resource or on a host that can manage the resource through a remote administration API.

DAML is a simple XML schema definition that enables the encoding of identity information in the form of an XML document so that it can be easily shared via IP protocols such as HTTP/S, as shown in Figure 3-5.

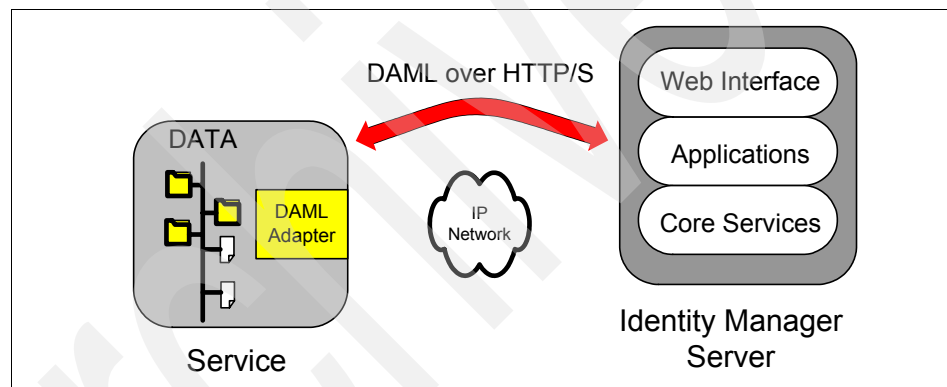


Figure 3-5 DAML connectivity to a service

Transactions from the Tivoli Identity Manager server are sent securely via HTTPS to the service adapter and then processed by the adapter. For example, if a service has just been connected to the Tivoli Identity Manager server, the accounts that already exist on the server may be reconciled or pulled back in order to import the users' details into the Tivoli Identity Manager LDAP directory. If a password change or a provisioning of a new user occurs, the information is transferred to and then processed by the adapter. The adapter deposits the new information within the application or operating system that is managed.

3.2 Physical component architecture

An Tivoli Identity Manager system is always deployed as part of an enterprise environment. A goal of the physical component architecture is to be flexible enough to support different configuration options. This section discusses the different network zones that you find within an enterprise environment and the placement options for the different Tivoli Identity Manager components.

3.2.1 Component configuration and placement

It is possible to deploy Tivoli Identity Manager components within a single network. While this kind of architecture may be reasonable for a lab or development environment, it is generally not recommended for a production setting.

We discuss how various Tivoli Identity Manager components relate to the network configuration and provide recommendations for how they should be distributed in a typical architecture.

3.2.2 Network zones

We must consider four types of network zones in our discussion of Tivoli Identity Manager component placement:

- ▶ Uncontrolled (the Internet)
- ▶ Controlled (an Internet-facing DMZ and the intranet)
- ▶ Restricted (a production network)
- ▶ Secure (a management network)

Since we do not place any components in an uncontrolled zone, we take a closer look at the remaining zones.

Internet DMZ (controlled zone)

The Internet demilitarized zone (DMZ) is generally a controlled zone that contains components with which clients may directly communicate. It provides a *buffer* between the uncontrolled Internet and internal networks. Because this DMZ is typically bounded by two firewalls, there is an opportunity to control traffic at multiple levels:

- ▶ Incoming traffic from the Internet to hosts in the DMZ
- ▶ Outgoing traffic from hosts in the DMZ to the Internet
- ▶ Incoming traffic from internal networks to hosts in the DMZ
- ▶ Outgoing traffic from hosts in the DMZ to internal networks

As a typical Tivoli Identity Manager deployment would include integration with a Tivoli Access Manager environment, we must consider the placement of other components such as WebSEAL, which could be used to protect access to the HTTP server used by the Tivoli Identity Manager GUI server. The DMZ is an appropriate location for the WebSEAL component of Tivoli Access Manager, and in conjunction with the available network traffic controls provided by the bounding firewalls, it provides the ability to deploy a highly secure Web presence without directly exposing components that may be subject to attack by network clients.

Note: The latest information about IBM Tivoli Access Manager can be obtained from *Enterprise Business Portals with IBM Tivoli Access Manager*, SG24-6556, *Enterprise Business Portals II with IBM Tivoli Access Manager*, SG24-6885, and *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014.

Production or management DMZs (restricted/secure zones)

One or more network zones may be designated as *restricted* or *secure*. That is, they support functions to which access must be strictly controlled, and direct access from an uncontrolled network should not be permitted. As with an Internet DMZ, a restricted network is typically bounded by one or more firewalls and incoming/outgoing traffic may be filtered as appropriate. Access to a secure zone is only available to a small group of authorized staff. Access into one area does not necessarily give you access to another secured area.

These zones typically would contain Tivoli Identity Manager server components and Tivoli Access Manager back-end servers that do not directly interact with users.

Intranet (controlled zone)

Typically, a controlled zone, such as a corporate intranet behind one or more firewalls, is not heavily restricted in use, but an appropriate span of control exists to ensure that network traffic does not compromise the operation of critical business functions.

You might need to place certain Tivoli Identity Manager components, such as the database server or directory server, in the intranet network to maximize the performance of data throughput or the availability of certain components or applications. In such cases, ensure that you do not compromise security in accessing these components or in the data flow between the components.

Other networks

Keep in mind that the network examples that we use do not necessarily include all possible situations. There are organizations that extensively segment functions into various networks. However, in general, the principles discussed here may be easily translated into appropriate architectures for such environments.

Placement of various Tivoli Identity Manager components within network zones is, on the one hand, a reflection of the security requirements in play, and on the other, a choice based upon an existing/planned network infrastructure and levels of trust among the computing components within the organization. While requirement issues may often be complex, especially with regard to the specific behavior of certain applications, determination of a Tivoli Identity Manager architecture that appropriately places key components is generally not difficult. With a bit of knowledge about the organization's network environment and its security policies, reasonable component placements are usually easily identifiable.

Figure 3-6 summarizes the general Tivoli Identity Manager component type relationships to the network zones discussed above.

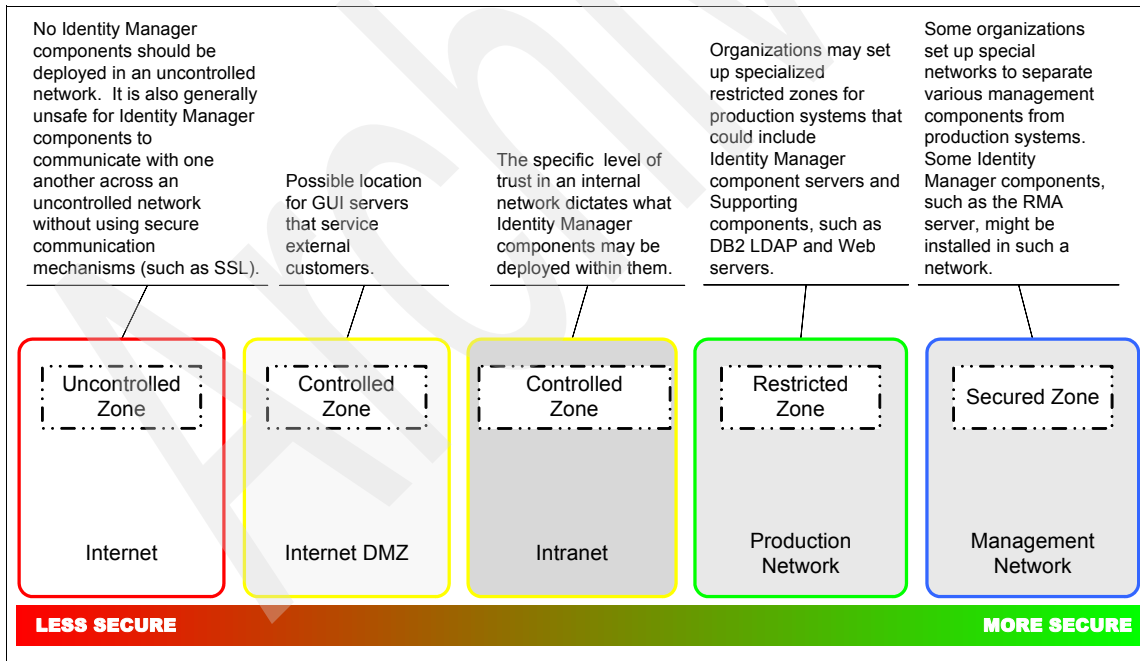


Figure 3-6 Network zones for Tivoli Identity Manager placement

As all the components of Tivoli Identity Manager have either information that access should be restricted to, or support such resources, we recommend that they all be placed in a restricted zone. An exception to this may be to place a Web server in the DMZ to manage external requests from business partners or customers if no general access control solution, such as Tivoli Access Manager WebSEAL, is in place.

3.2.3 Integrating with Tivoli Access Manager

When integrating Tivoli Identity Manager with Tivoli Access Manager, we must consider the placement of other components, such as the Tivoli Access Manager Policy Server and the WebSEAL server. WebSEAL would be used to protect the Web server associated with the Tivoli Identity Manager server, as well as the other corporate Web servers.

As depicted in Figure 3-7, we recommend that WebSEAL servers accessible via the Internet should be placed in a DMZ. WebSEAL in such a setting should generally be in a network zone separate from those that contain other Tivoli Access Manager components upon which it relies, and from the Web servers to which it is junctioned. In this case, it is not necessary to place the Web servers that service external users in the DMZ, as the WebSEAL server in the DMZ acts as a proxy to the Web server, which should be placed in a restricted zone. The Tivoli Access Manager Policy Server may be placed in the controlled intranet zone if there is sufficient trust in those who can access that zone, and communication is secure using SSL. However, in most cases, the most suitable place for the Tivoli Access Manager Policy Server would be in the secured zone. The Tivoli Access Manager User registry master should also be placed in the secured zone.

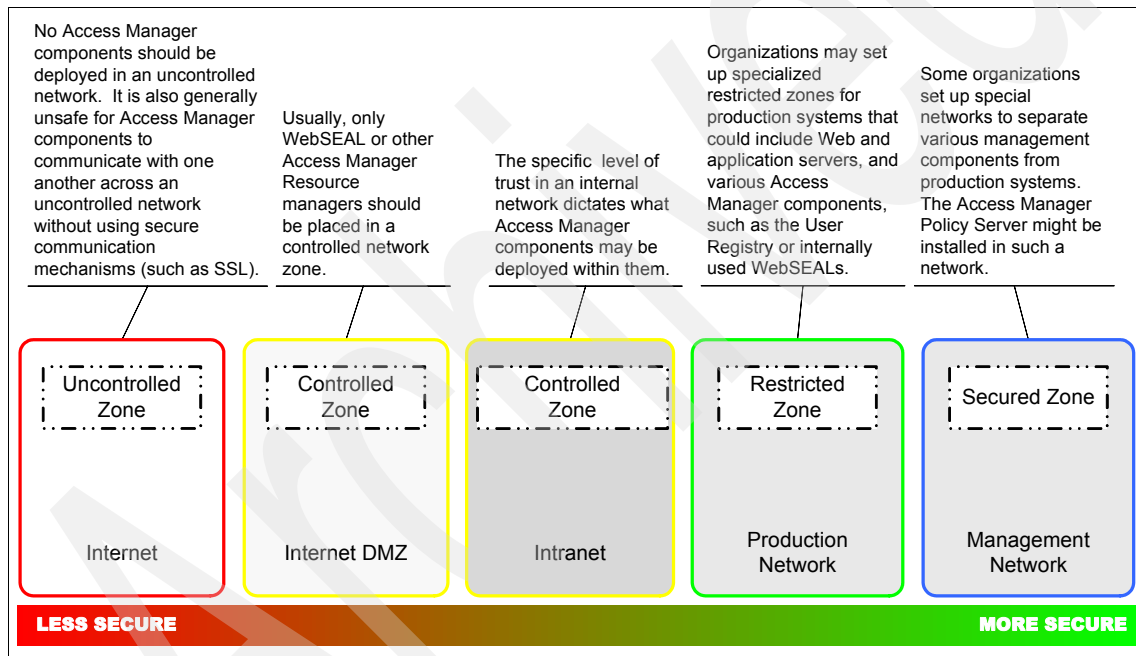


Figure 3-7 Network zones for Tivoli Access Manager placement

Figure 3-8 shows an example architecture for integrating Tivoli Identity Manager and Tivoli Access Manager. Note that firewalls are introduced to separate the networks and permit access only through specified ports. In this example, access from the Internet is only allowed on ports 80 and 443 to the WebSEAL server in the DMZ, and the WebSEAL server is configured to access the back-end Web servers through alternative ports, hence, forcing all users requesting access to the back-end Web servers to be authenticated by WebSEAL.

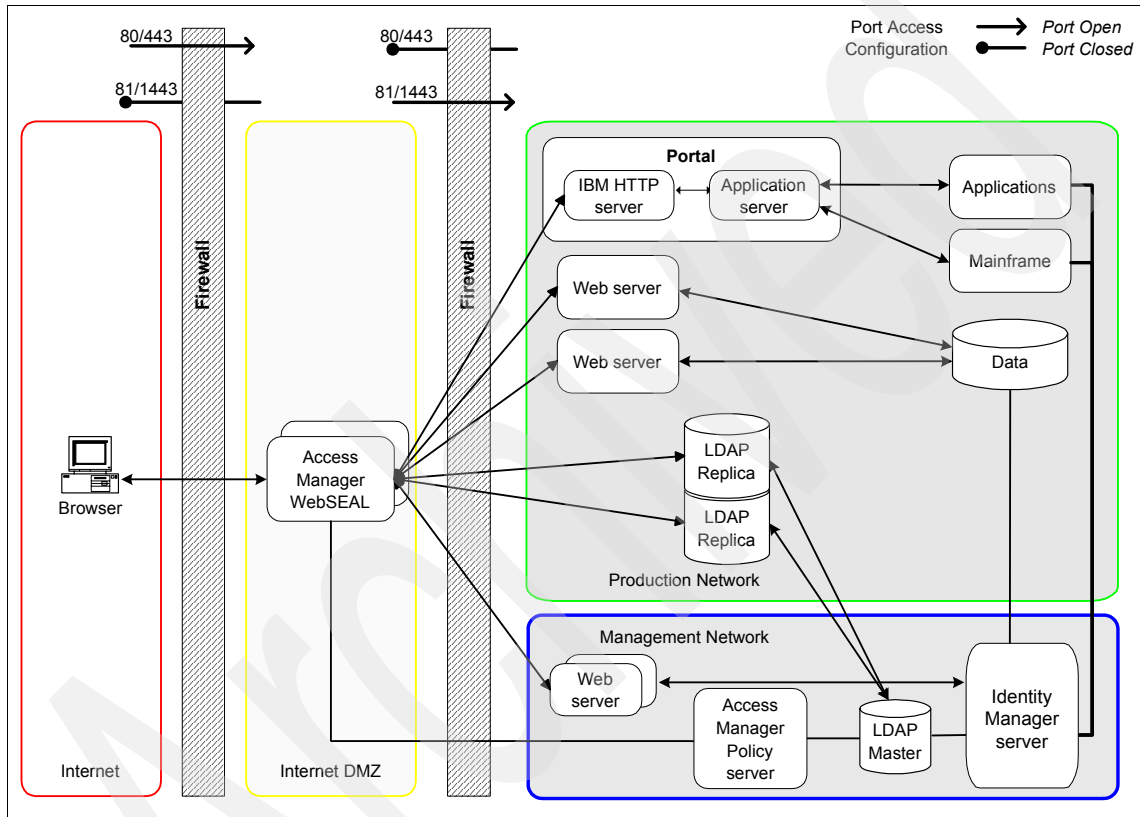


Figure 3-8 Integrated architecture for Tivoli Access Manager and Tivoli Identity Manager

Note: Port 80 is the default port for HTTP, and port 443 for HTTPS, through which the browser connects to the Tivoli Identity Manager GUI.

If LDAP is being used for the user repository, the firewall could be configured to allow access to the registry only via the WebSEAL server. More information about the integration with Tivoli Access Manager can be found in Chapter 6, “Tivoli Access Manager integration” on page 275.

Archived

Detailed component design

In this chapter we look at the design considerations and options for the main configurable components. The goal of this chapter is to give you the building blocks for a detailed Tivoli Identity Manager design.

The structure of the chapter is consistent with Chapter 3, “Tivoli Identity Manager component structure” on page 83, and covers:

- ▶ Tivoli Identity Manager entities
- ▶ Tivoli Identity Manager management entities
- ▶ Tivoli Identity Manager functions
- ▶ Tivoli Identity Manager user interface and access control
- ▶ Tivoli Identity Manager schedules
- ▶ Adapter connectivity
- ▶ Software and hardware requirements

In each section of this chapter we take a consistent approach of breaking each component into bite-sized chunks and then, for each chunk, looking at the technical aspects of it (*how it works*) before examining any design considerations (*how can you make it do what you need for this deployment*). We close each section with a summary that brings all the bite-sized chunks back together from a design perspective.

4.1 Tivoli Identity Manager entities

Tivoli Identity Manager is concerned with managing users and their accounts. Passwords, group memberships, and other attributes are associated with the users and accounts. These 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, Tivoli Identity Manager groups and access control items (ACIs), and policies. Tivoli Identity Manager also contains workflow, audit logs, and reports. These are described in the following sections.

The entities managed by Tivoli Identity Manager are:

- ▶ Users, accounts, services, and attributes
- ▶ Passwords
- ▶ Group memberships
- ▶ Accesses
- ▶ Managed systems and applications

4.1.1 Users, accounts, services, and attributes

A person can be classified as a *person*, business partner person (*BPPerson*), or *custom Person*. A person is typically an employee of the company or organization. A *BPPerson* is typically an individual who needs access to an organization's 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 must be extended for the organization.

Note: In most cases, we recommend that you create a *custom person*. This allows you to modify the person in the future. This is discussed in more detail in 4.6.8, “Custom person classes” on page 213.

A person can be located anywhere in the *organization tree*, so the organization tree represents the user structure of a company.

The personal information is defined as attributes on the person objects. This may include first, last, and full names, phone numbers, employee number, supervisor, and e-mail address.

The person corresponds to the *inetOrgPerson* object, and not the *ePerson* object (although the object classes are similar).

An *account* is a person's access to Tivoli Identity Manager or to a service (managed resource), such as Active Directory, Solaris, SAP, and so on. Accounts have attributes that are defined by the managed resource's corresponding service.

Services represent an instantiation of a specific platform type that hosts account information. For example, a service type may be *LDAP profile* or *Active Directory profile*. A service of type Active Directory profile would need to be configured with the appropriate information to access a specific Active Directory's account repository, in order to collect information about its accounts and start managing them. The necessary information to set up the service typically includes:

- ▶ A path to the managed resource's adapter (An adapter relays account management operations between the Tivoli Identity Manager server and the managed resource.)
- ▶ Some configuration information specific to the platform, such as a set of credentials that are used to access the managed resource's account information.

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. Accounts can also be orphaned manually through the Tivoli Identity Manager user interface.

4.1.2 Passwords

All accounts have passwords. Account passwords can be centrally managed by their owners or administrators using the Tivoli Identity Manager Web interface.

Password management is a very important topic. Since passwords represent access to corporate applications, they must be securely managed during their entire life cycle. Tivoli Identity Manager provides a full set of features to manage the passwords in a secure environment.

There are two options for password management within Tivoli Identity Manager: Passwords can be synchronized or not. We define *password synchronization* as a process or technology that maintains a single password that is subject to a single password policy, and changes on a single schedule across multiple systems.

The synchronization is applied to *all accounts* associated with the user. 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 exceptions to this are:

- ▶ The password synchronization plug-in for Windows Activity Directory servers
- ▶ Password synchronization plug-in for i5/OS®
- ▶ Password synchronization adapter for the Tivoli Access Manager, which intercepts a password change on the managed platform and passes it through Tivoli Identity Manager

When the password synchronization property is enabled, there is only one global password for all the applications managed by Tivoli Identity Manager. If an account is being set up for the first time, password synchronization does not apply. There is only one account, and therefore, one password.

If a user has more than one account, password synchronization affects the following user or administrator actions:

- ▶ Creating a new account
- ▶ Changing a password for an existing account
- ▶ Provisioning an account
- ▶ Resetting an expired or forgotten password for an existing account
- ▶ Restoring an account that was suspended

If you have enabled the password synchronization property, there is no way for a user to change the password of only one account. All accounts receive the password change. Without the password synchronization option enabled, users could select which accounts are to be changed. Administrators can always change passwords for selected accounts by using the service's native account management facility, but this would imply that a user will have different passwords across platforms or applications, because the reconciliation process does not synchronize passwords.

There is a process where Tivoli Identity Manager generates a random password. This can be displayed to an administrator or mailed to a user. Also, there is the option where Tivoli Identity Manager could generate a password for an account and send a URL to the user for password pickup using the shared secret attribute for password pickup.

Tivoli Identity Manager uses a challenge/response function to verify a user's identity if they have forgotten their Tivoli Identity Manager password. The challenge questions can be picked from an administrator predefined list or defined by the user. When a user logs into Tivoli Identity Manager for the first time, he enters or selects the challenge questions (if configured) and responses. On subsequent logins to Tivoli Identity Manager, he can select a *forgot your password* option, and then a subset of the challenge responses are used to verify the user.

4.1.3 Group membership

Accounts are assigned privileges on target systems and applications via some form of group membership. These may be groups on UNIX systems or Windows domains, SAP groups or profiles, or another access control grouping mechanism. Membership is granted by using a group attribute on accounts.

Group lists, for most managed targets, are updated with the reconciliation function. Thus, administrators do not manually enter group names. They select from a list that is in synch with the respective target.

By default, Tivoli Identity Manager manages group memberships, like other account attributes, via its assigned account management process. For example, adding or removing a group assignment to an account in Tivoli Identity Manager is handled in the same way as an attribute change, by accessing the account management window, as shown in Figure 4-1, and modifying the group attribute's values. User permissions to view and modify a group are enforced at the attribute level, and apply for all values.

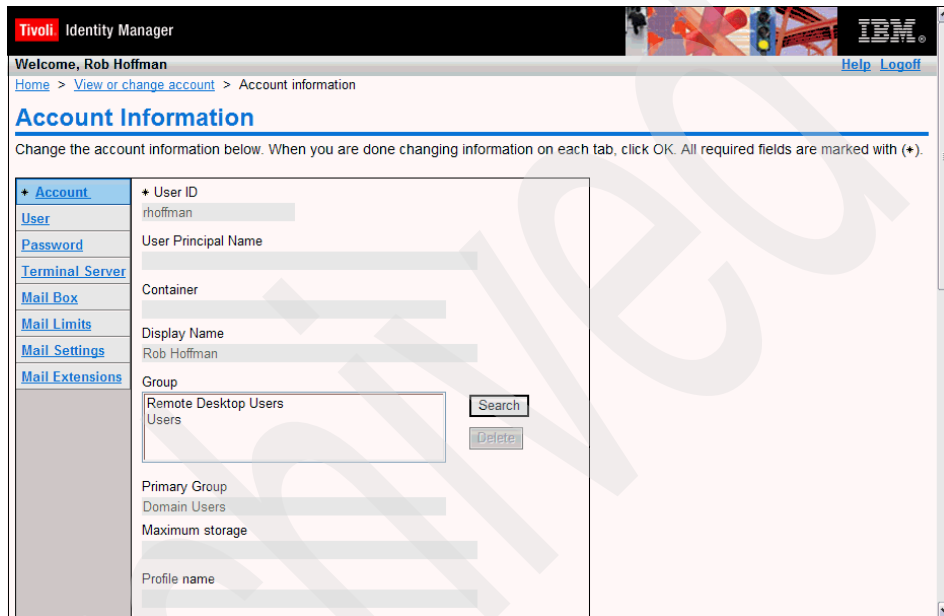


Figure 4-1 Group management in Tivoli Identity Manager's self-care interface

Tivoli Identity Manager supports the creation and deletion of groups by way of the Tivoli Identity Manager administration interface, if the adapter supports this feature (see the adapter configuration guides¹ for details). However, Tivoli Identity Manager will not manage ACLs, such as file permissions or other resources, on managed targets. This task must be performed by the local administrators or application owners using the native system or application tools.

4.1.4 Accesses

Accesses are manual permissions that can be requested for Tivoli Identity Manager users in both the administrative console and the self-service interface.

¹ The adapter configuration guides can be found in the Tivoli Identity Manager Version 5.1 Information Center at the following address:

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

Accesses can be defined for both groups and Tivoli Identity Manager organizational roles.

Groups defined as accesses are assigned a configurable access type, and can be assigned an owner, an access workflow, an access category and quick access options, an alternative name and description, and notification options. They can also be assigned a recertification policy, and can be made available for users to request directly via a separate menu, rather than through the account editing process.

Roles can also be defined as Tivoli Identity Manager Accesses. However, the only configuration options available to organizational roles defined as accesses are an access category and a quick access option. There is no option to assign a workflow to a role defined as an access, or define an alternative name or description. Only static roles configured as accesses can be requested. Dynamic roles configured as accesses are automatically assigned to users, and cannot be requested, only viewed as accesses granted to a user.

The access request functionality provides groups with user-friendly categorization, names, and descriptions, as well as a dedicated workflow process when requested, and additional notification options. Only accesses assigned to compliant groups are available to users, so that if a user does not have a policy granting him access to an account, the corresponding access will not be available to him. A typical access request using Tivoli Identity Manager's self-care interface is depicted in Figure 4-2.

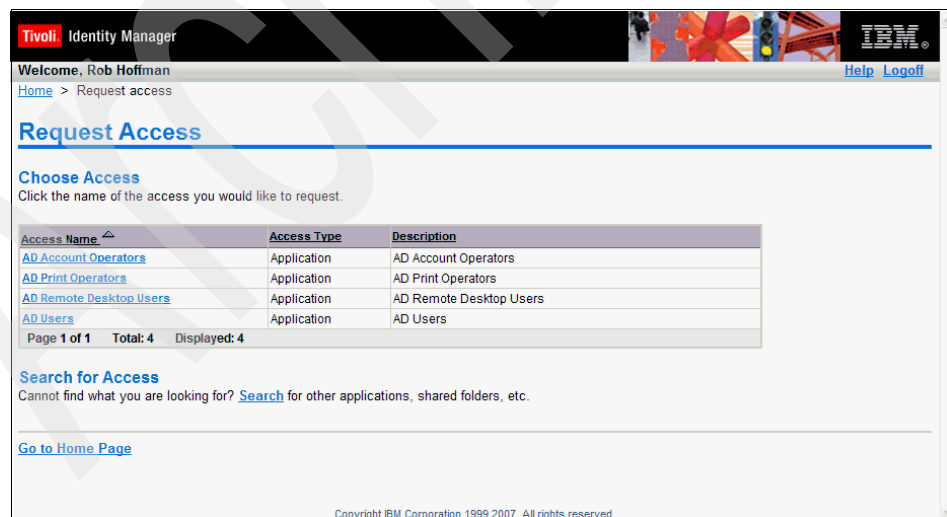


Figure 4-2 Access request in Tivoli Identity Manager's self-care interface

The following access types are configured by default in Tivoli Identity Manager:

- ▶ Role
- ▶ Application
- ▶ Shared folder
- ▶ E-mail group

However, these access types can be further configured in Tivoli Identity Manager to suite business requirements.

4.1.5 Managed systems and applications

Tivoli Identity Manager manages users on many managed systems. These include operating systems, such as many versions of UNIX and Windows servers, and applications, such as databases and business applications. Tivoli Identity Manager can perform user management requests either by communicating with the managed resources or by notifying a managed resource owner to perform the request manually.

Interfacing directly with managed systems

In order to interact with a managed resource platform, Tivoli Identity Manager deploys an adapter to perform the administration of accounts on the system or application. Some adapters are deployed to the system or application and interact locally. Others can operate remotely and be deployed anywhere in the network, and communicate with multiple instances of the same managed resource type.

There are two main types of adapters:

- ▶ DAML-based adapters (also known as ADK-based adapters or out-of-the-box adapters)
- ▶ RMI adapters

DAML and RMI both refer to the protocol used by these two types of adapters to communicate with the Tivoli Identity Manager server: Directory Access Markup Language and Remote Method Invocation. DAML-based adapters are platform-specific components dedicated to communicating with the managed target type, whereas RMI-based adapters use Tivoli Directory Integrator as a platform to interact with the managed resources. RMI-based adapters can be customized to behave differently using the Tivoli Directory Integrator framework.

For more information about Tivoli Directory Integrator, refer to the IBM Redbooks publication *Robust Data Synchronization with IBM Tivoli Directory Integrator*, SG24-6164, and the IBM Tivoli Directory Integrator Information Center (for Version 6.1.1 as well as Version 7.0) at:

<http://www.ibm.com/developerworks/wikis/display/tivolidoccentral/Tivoli+Directory+Integrator>

For both types of adapters, a set of definition files is included with each adapter that must be imported into Tivoli Identity Manager. This set of definitions is called a *service profile*, and it provides Tivoli Identity Manager with group and account specifications for the managed resource, and in the case of RMI-based adapters, account management process definitions as well.

Adapter-to-server communication by default does not use Secure Sockets Layer (SSL). However, SSL communication can be enabled (both one-way and two-way SSL). While the procedures are different for DAML-based and RMI-based adapters, each procedure is documented in the adapter's documentation. These procedures involve configurations to be performed both on the application server and on the adapter side.

Let us use an example on the Active Directory platform, with which Tivoli Identity Manager communicates using a DAML-based adapter. On the client side you must run the adapter configuration by executing **agentCfg -agent <adapter>Agent** from the install bin directory. In the protocol section of the adapter configuration window, the default value for use_SSL is false. For using SSL you also must configure the use of digital certificates by running **certTool -agent <adapter>Agent** from the install bin directory.

On the Tivoli Identity Manager Server, SSL is configured via the WebSphere Application Server console. For more information, there is a full explanation in the section on "Secure communication with supported middleware" of the Tivoli Identity Manager Version 5.1 Information Center.

Manual account management

RMI-based adapters provide a framework that can help you create new adapters for managed resources if an Tivoli Identity Manager adapter is not available. There may be situations in which it is not possible or it is impractical to interface with a managed resource. This may include situations where:

- ▶ No application programming interface (API) is available for the managed resource.
- ▶ A person is required to perform the account management operations on the managed resource.

- ▶ It is easier to perform the account management operations manually on the managed resource.

Tivoli Identity Manager provides the possibility to manage users on managed resources by notifying the resource's owner or administrator of account management requests. The resource's owner or administrator can then perform the account management request and notify Tivoli Identity Manager of the result of the operation. This is known as a manual service within Tivoli Identity Manager.

Services and service types

Each adapter instance and, by extension, managed resource type is defined as a *service type* within the Tivoli Identity Manager server. Accounts are associated with specific services, which represent an instance of the corresponding managed resource type or service type. For example, there is a *service type* representing the HPUX platform, and a *service* for every HPUX server. The services are defined within the organization tree and can have ACIs attached to control administrative access to functions performed against the service. A service can only be defined for a pre-existing service type.

It is possible to define account defaults for a configured service or service type. These account defaults are preconfigured attribute values, either static or dynamically created based on the account owner's attributes. For example, an account default for a Windows Active Directory service or service type may assign the person's full name to the account's display name attribute. However, the account requester may change the attribute value if they wish to.

4.2 Tivoli Identity Manager management entities

Tivoli Identity Manager uses the following entities for management:

- ▶ Organizational tree
- ▶ Organizational roles
- ▶ Tivoli Identity Manager groups and ACIs
- ▶ Accesses
- ▶ Policy
- ▶ Account defaults
- ▶ Workflow
- ▶ Audit logs
- ▶ Reports
- ▶ Life cycle management

These are discussed in the following sections.

4.2.1 Organizational tree

The Tivoli Identity Manager organizational tree, also known as the orgtree, defines the structure for the organization into which Tivoli Identity Manager is being deployed. The tree consists of:

- ▶ An organization: There is normally only one organization at the top of the org tree, although it is possible to create additional ones at the top of the tree.
- ▶ One or more locations: These are locations defined by the business.
- ▶ One or more organizational units: These are teams or departments as defined by the business.
- ▶ One or more business partner organizations: These are business partners as defined by the business.
- ▶ One or more admin domains: These are Tivoli Identity Manager groupings for administration.

The Tivoli Austin Airlines organization consists of a number of region-based customer service locations (such as center region and east region), head office, business partners, and an AD_Corp admin domain. Each of these contains further sub region organizational units, such as the Denver and Raleigh customer service locations, as depicted in Figure 4-3.

Note: The Tivoli Austin Airlines organization is later used as our fictive enterprise when we discuss a real-life scenario for deploying Tivoli Identity Manager in Part 2, “Customer environment” on page 297.

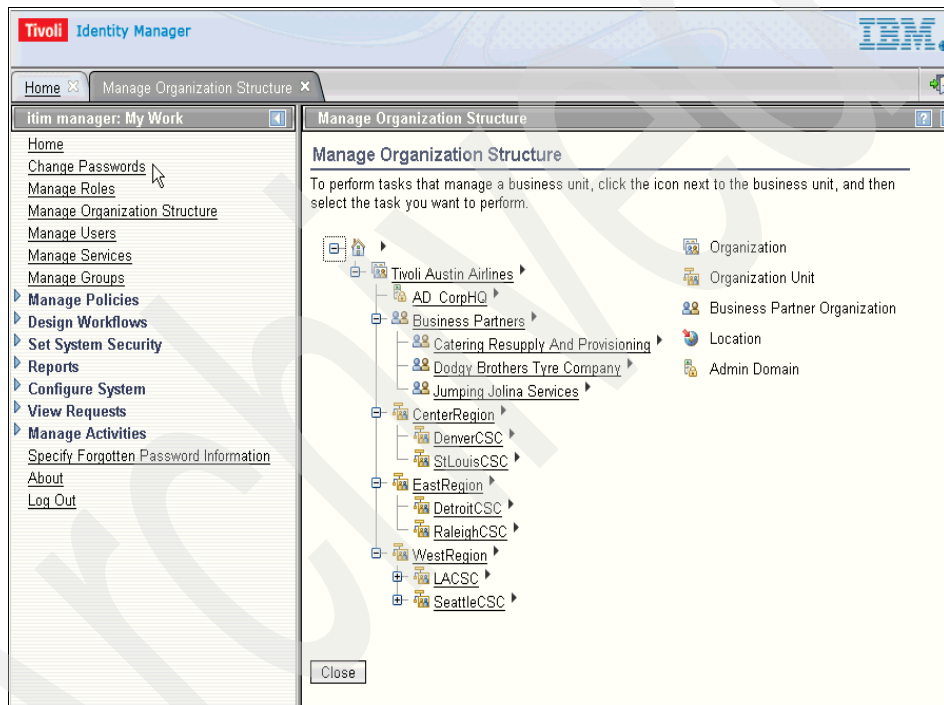


Figure 4-3 Tivoli Identity Manager organization tree of Tivoli Austin Airlines

The arrow symbol at the end of each node, when clicked, extends a context menu that offers a number of options. These options typically include the possibility to change the properties of the corresponding container or add a new container below the current one.

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 wish.

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

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

Tivoli Identity Manager organizational roles and ACIs are also attached to points in the org tree.

4.2.2 Organizational roles

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. Organizational roles can be *static* or *dynamic*. In static organizational roles, assigning a person a static role is a manual process, and it can be done every time it is needed, during an identity creation or through an identity feed.

Dynamic organizational roles set person membership to a specific role based on valid Lightweight Directory Access Protocol (LDAP) filters. Dynamic organizational roles are evaluated at different times:

- ▶ When a new person is created in the Tivoli Identity Manager system
- ▶ When a person's information changes in the Tivoli Identity Manager system
- ▶ When a new dynamic organizational role is created

Every time that a dynamic organizational role is evaluated, all people who fit the LDAP filter affected with the membership of the role and their personal information is updated with the membership information.

Roles can also be configured as Tivoli Identity Manager *accesses*. Organizational roles set up as accesses can be assigned an access category and quick access options. Static organizational roles configured as accesses can be viewed and requested by users. Dynamic organizational roles configured as accesses can only be viewed by users because members are dynamically assigned.

Role hierarchy

Roles can be organized into *role hierarchies*, which allow the administrator to plan and build hierarchical role structures and implement *role relationships*. Child roles inherit all privileges from their ancestors. Roles may also have classifications assigned; by default, these are *application roles* or *business roles*. Additional role classifications may be added by updating the Tivoli Identity Manager configuration file `enRole.properties`.

Roles may be assigned owners, which can be a person or another role. The benefit of *roles owning roles* is useful when role membership approval has been configured.

Figure 4-4 shows a simple role hierarchy. In this diagram, the child roles inherit from its ancestors. Users defined as members of the manufacturing or headquarters divisions are child roles of the employee role. Both the employee role and manager role are child roles of WW VPN access. When a new employee is added to the manufacturing division, they automatically inherit the accesses and privileges of the WW VPN access role.

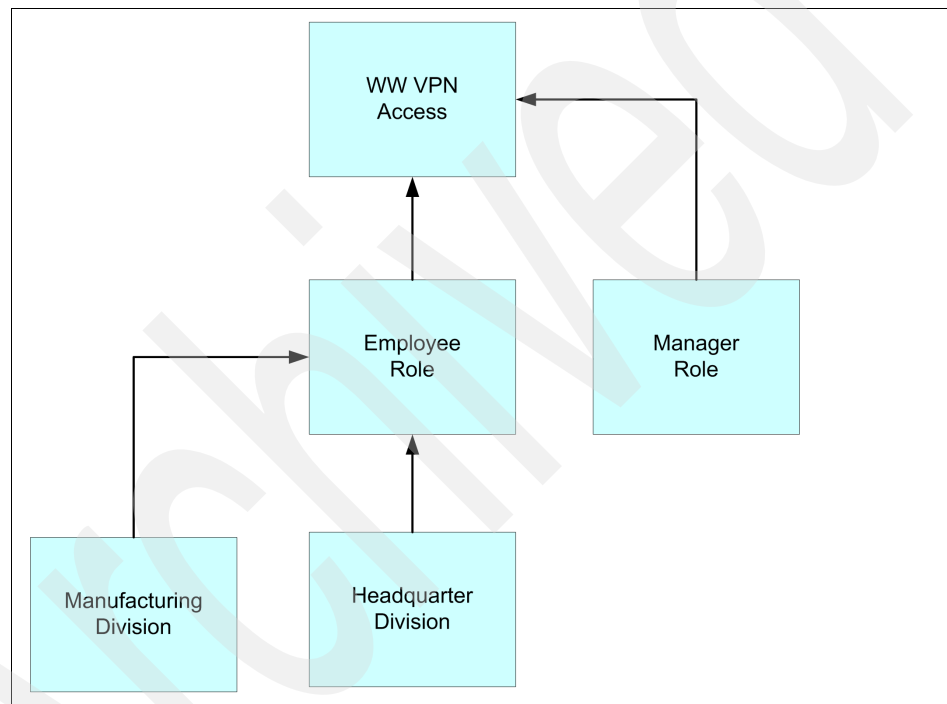


Figure 4-4 Simple block diagram of role hierarchy

Note: Only static organizational roles may be part of role hierarchies.

To manage the role hierarchy within the administration interface, select the option **Manage Roles**. Search for the role you wish to manage. Once the roles are listed in the interface, click the twisty next to the role name to expose the extended management options, such as Manage Child Roles or Add Child Role. See Figure 4-5 on page 115 for an example.

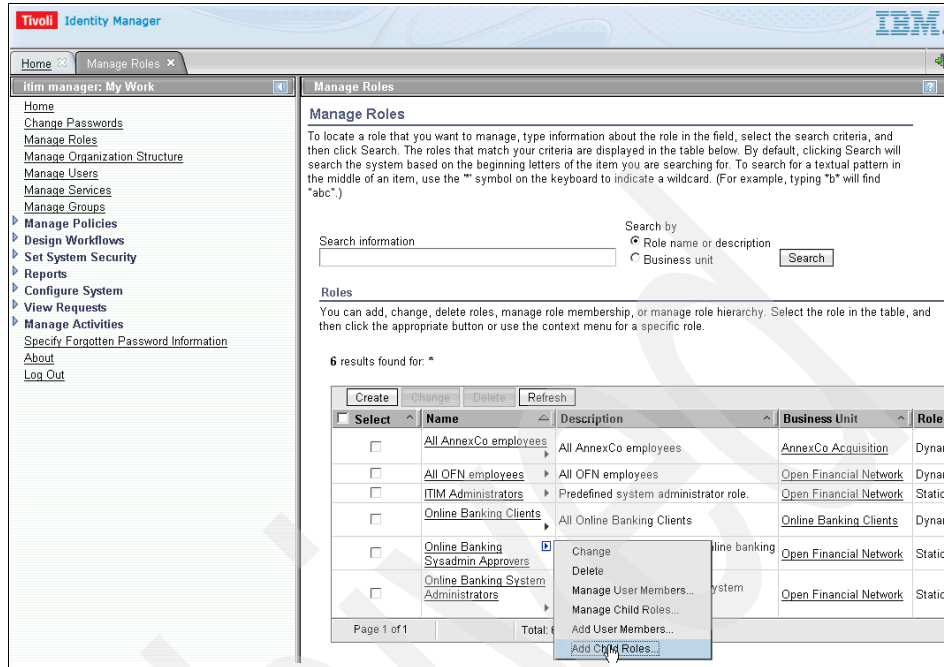


Figure 4-5 Manage Roles: Add Child Role

Role approvals

Identity Manager provides the means to configure *role approvals*. This is not provided in the base configuration, but it is available by way of workflow customization. The documentation provided with your Tivoli Identity Manager installation describes the necessary steps. Using a Web browser, open the following file:

```
file:///<ITIM_HOME>/extensions/5.1/examples/workflow/roleApproval/index.html
```

Once role approval is implemented, when a user requests a role or a role is assigned by an administrator or other automated means, such as an identity feed, role approval workflow will be triggered. An approval activity and notification e-mail will be routed to the role owner(s). If approved, then the user will become a member of the requested role.

When creating or modifying roles in Identity Manager, you may expose the role to access, and if required, common access. This allows users to request access to roles by way of the self-service interface using the Request Access link.

Figure 4-6 on page 116 shows the manage role interface; note that the *Enable access for this role* check box is checked. To learn more about access review, refer to 4.1.4, “Accesses” on page 106.

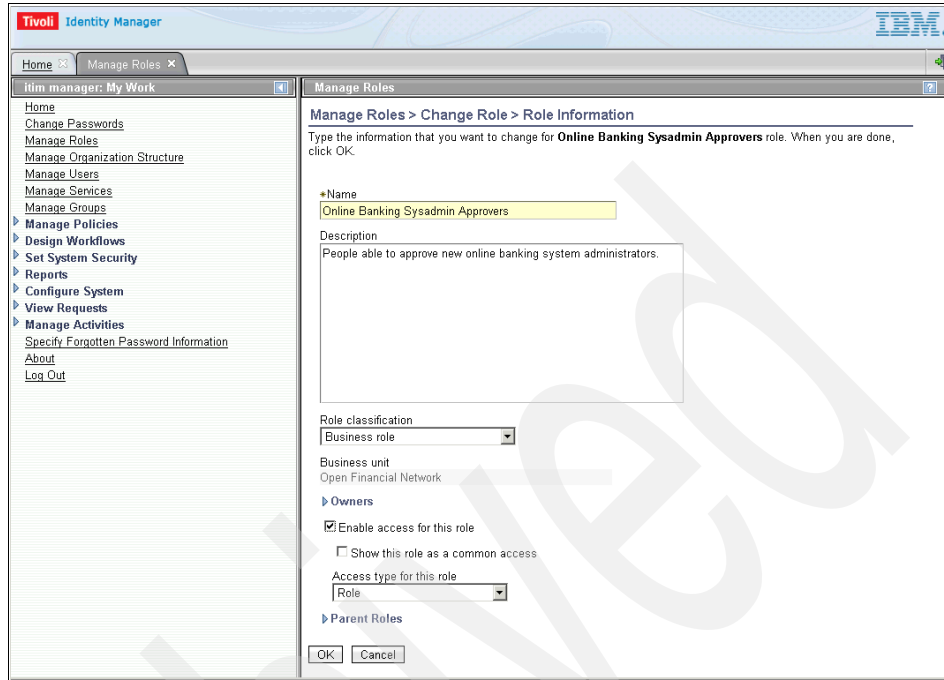


Figure 4-6 Change role with access enabled

Organizational role recertification

Role membership may be recertified as part of a recertification policy, which targets the object type of user. You may select all roles or selected roles. Refer to the recertification policy discussion in 4.2.10, “Life cycle management” on page 129 and 4.3.4, “Apply policies to person and account management” on page 141.

4.2.3 Tivoli Identity Manager groups and ACIs

A user's access within Tivoli Identity Manager, for example, the functions that they can perform in Tivoli Identity Manager, is governed by the groups to which they belong. Tivoli Identity Manager governs user access rights using an Access Control Item (ACI). An ACI controls user access by defining the access privileges of an Tivoli Identity Manager group or ACI principal. A principal is an Tivoli Identity Manager user that has a defined relationship with the object targeted by the ACI.

Examples of a principal include:

- ▶ A person's manager (for a person)
- ▶ A sponsor (for a business partner person)
- ▶ The user himself (for his own accounts)
- ▶ The owner (for a role or service)
- ▶ The owner of any access defined for a service (for the service for which the access has been defined)
- ▶ All users

Members of an Tivoli Identity Manager group or ACI principal can view and perform operations on attributes within a target class (context) as defined by the scope of the ACI.

Tivoli Identity Manager groups also define which Tivoli Identity Manager interface views to which they have access.

Note: Only Tivoli Identity Manager groups or principals can be assigned to an ACI. Organizational roles cannot be assigned to an ACI. ACIs grant or deny the ability to perform Tivoli Identity Manager functions.

This role-based access is for Tivoli Identity Manager users assigned to the Tivoli Identity Manager groups. Members of the Tivoli Identity Manager system administrators group 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.

4.2.4 Policy

Tivoli Identity Manager employs seven types of policy:

- ▶ Provisioning policy
- ▶ Password policy
- ▶ Identity policy
- ▶ Service selection policy
- ▶ Adoption policy
- ▶ Recertification policy
- ▶ Separation of duty policy

Provisioning policy

A *provisioning policy* confers access to many types of *managed services* (Tivoli Identity Manager, Windows 2003, Solaris, and so on) by granting a person access based on an organization (for example, an organizational role). In other words, access to a target-managed service is either:

- ▶ Granted to all persons in an organization.
- ▶ Granted only to persons assigned to a specified organizational role, or any organizational role that is a child of the specified organizational role, which forms part of a role hierarchy.
- ▶ 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 what accounts can be created for a user and it can, optionally, create the accounts on those systems as soon as one of the organizational roles governing the policy is assigned to the person, or when the provisioning policy's configuration has been updated. This is referred to as an *automatic provisioning policy*. It can also be used to define a specific approval workflow process that must be applied to the accounts.

At the time of creating a provisioning policy, there are many accounts that could be affected by this new policy. In a test environment, it might not have major implications, but, in a production environment, it could have implications based on access granted because of the policy. To prevent any accidental behavior of a provisioning policy there is a simulation function, called *policy preview*, that helps you understand what and who is affected by the creation or modification of a provisioning policy.

Provisioning policies take precedence over *account defaults*. Account defaults configured for services and service types will be overridden by provisioning policy entitlements.

Service selection policy

A *service selection policy* extends the ability of provisioning policies by provisioning a specific instance of a service 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 a JavaScript. For example, if your company operates in multiple geographical locations, a user is granted access to a print service depending on his personal location attribute.

Identity policy

An *identity policy* defines how a user's ID is created. Tivoli Identity Manager automatically generates user IDs from 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. 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.

Password policy

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:

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

Adoption policy

An *adoption policy* defines how a user account with no defined owner can be assigned one in Tivoli Identity Manager. When account information is obtained from a managed resource via a reconciliation, it is by default not associated with any person record in Tivoli Identity Manager. These accounts are called *orphan accounts* and cannot be managed by Tivoli Identity Manager, since permissions cannot be derived from person record information. Adoption policies define logic associating an account to a person record automatically during a reconciliation.

By default, accounts are associated to person records by matching the account's user identifier to the person record's aliases.

Adoption policies can be defined:

- ▶ At global level (for all services and service types)
- ▶ At service type level
- ▶ At service level

Adoption rules defined at the service level take precedence over those defined at the service type level, which take precedence over those defined at the global level.

Recertification policy

A *recertification policy* allows accounts, accesses, and users to be validated periodically, either at fixed dates or on a rolling calendar basis (for example, three months since the last time the item was recertified). A recertification policy is created by creating a workflow and assigning it to a calendar event.

Note: Recertification policies are not to be confused with life cycle rules, though they share a number of similarities. Life cycle rules are workflows, which can be executed periodically, but that are not dedicated to revalidation. Recertification policies are workflows dedicated to revalidating accounts, accesses, and users, either periodically or on a rolling calendar basis.

The recertification status can be viewed in the Tivoli Identity Manager interface and reports.

Separation of duty policy

A *separation of duty policy* allows you to manage rules that define which static organizational roles are mutually exclusive. In simple terms, members of role A cannot be members of role B and vice versa. Separation of duty (SoD) policies are used to protect against conflicts of interest. Separation of duty policies may be owned by a person or an organizational role.

Note: Currently, Tivoli Identity Manager only supports static roles with separation of duty policies.

More complex separation of duty policies can be defined. This may be used to ensure that no one person can control an entire process. Rules can be created so that a person can only be a member of a defined number of roles within the policy. For example, a separation of duty policy may contain roles A, B, C, and D. The rule could restrict membership to just three of the defined roles. So in our example, a user could have any three of those four roles, such as A, B, and D, but having all four of them would be a policy violation.

To identify conflicts of interest within your business, you need to understand the operational process flows and roles within these process flows. One technique would be to create flow diagrams and a role matrix. The flow diagrams should illustrate the data and functional process flows within the business, while the matrix, which should list each role identified on both the *x and y axis*, can be used to plot conflicting roles. The classic example of conflicting roles is invoicing and accounts payable, which could potentially enable someone to create an invoice and pay themselves.

Finally, before creating a SoD policy, you should conduct a risk assessment. Implementing your SoD policy could have an impact on your business, for example, do you have enough staff to implement this policy. In 1.1, “Security policies, risk, due care, and due diligence” on page 6, we review risk assessments.

Once you have implemented SoD policies, you need to consider how to handle violations and assign a policy owner for each SoD policy. A policy owner can grant exemptions to policy violations and document reasons for doing so.

Note: Exemptions should be regarded as a temporary solution to cover situations like staff shortages due to illness and so on. It is a best practice to ensure that all separation of duty policy violations are subject to regular review.

4.2.5 Account defaults

Account defaults can be specified for each Tivoli Identity Manager configured service or service type. These defaults are usually configured by mapping a person record field to an account field (for example, a person record's full name field to an account's display name field). Alternatively, it is possible to use JavaScript to define more complex mappings.

Account defaults differ from provisioning policy in that account defaults do not define the set of users who are allowed to have accounts or what attribute values are compliant, but rather defines only the default values for a new account if a user can have it. The change of account default configuration does not affect the compliance status of user accounts.

The following list highlights the differences between the use of account defaults and provisioning policies:

- ▶ Like *default* provisioning parameters, account defaults specify the default values for account attributes during provisioning.
- ▶ Unlike default provisioning parameters, account defaults are not scoped by membership. They apply to all users.
- ▶ Unlike default provisioning parameters, account defaults do not have implications on compliance. A value specified as an account default is not automatically treated as an allowed value.
- ▶ Values that are specified as account defaults do not appear within the entitlement parameter list. They are entirely independent from provisioning policy.

- ▶ Provisioning parameters take precedence over account defaults. Specifically, mandatory and default provisioning parameters will override an account default for the same attribute.

4.2.6 Workflow

A workflow is a set of steps or activities that define a business process. You can use the Tivoli Identity Manager workflows to customize account provisioning using:

- ▶ Account request workflows
- ▶ Access request workflows
- ▶ Operation workflows
- ▶ Life cycle rule workflows
- ▶ Recertification workflows

An *account request workflow* can be associated with a provisioning policy entitlement. Account request workflows define pre-conditions to account provisioning (such as Linux or directory accounts).

An *access request workflow* can be added to a service access. Access request workflows define pre-conditions to access provisioning (such as groups or shared drives).

Operation workflows define the actual sequence of activities taking place when performing provisioning operations, on person, business partner person, and account. They can be customized by editing the sequence of activities occurring during the operation, as well as the activities themselves. Tivoli Identity Manager life cycle rules use existing operational workflows.

Typical examples of access request workflows and account request workflows may include approvals and requests for information. Typical examples of operation workflows may include modifying a business process to include additional activities, such as modifying a person record attribute, sending an additional notification, or performing an action on another account.

Life cycle rule workflows are actually operation workflows that are associated with a life cycle rule. They are then executed for a define set of people or accounts, either periodically or manually. Typical examples of life cycle rules may include periodic password resets.

Recertification workflows are workflows created within recertification policies. They define which activities take place when accounts, accesses or users must be recertified. Typical examples of recertification workflows may include a recertification approval to be sent to an account owner, followed by either an account recertification or suspension, depending on the result of the approval.

Workflows are customized using the Tivoli Identity Manager administrative interface. For all workflows apart from operation workflows, two interfaces are proposed, a simple interface and an advanced interface. The simple interface lets the user select frequently used activities from a drop-down list, configure them, and list them in a simple sequence. The advanced interface launches a Java graphical interface, in which all available activities can be dragged-and-dropped and linked together to form workflows of varying complexity.

Operation workflows are only available via the advanced interface.

4.2.7 Logs and audit

Tivoli Identity Manager employs logging features that log the events during specific transactions. This facilitates isolating and debugging problems, focused on troubleshooting key Tivoli Identity Manager business processes, like:

- ▶ Add, modify, suspend, restore, and delete a person.
- ▶ Add, modify, suspend, restore, and delete an account.
- ▶ Add and remove access.
- ▶ Change a password.
- ▶ Add, modify, and delete a provisioning policy.
- ▶ Add, modify, and delete a dynamic role.
- ▶ Add, modify, and delete a service selection policy.
- ▶ Reconciliation and event processing (including identities).
- ▶ Add, modify, and delete a separation of duties policy.

There are several types of logging available, because Tivoli Identity Manager works with different applications:

- ▶ Installation log
- ▶ Audit trail (view requests) in the Web user interface
- ▶ Tivoli Identity Manager server log
- ▶ Web server access log
- ▶ Directory server log

To learn more about logs, refer to the Tivoli Identity Manager Version 5.1 Information Center, found at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_trouble_logs.html

Click **Troubleshooting and support**, then **Diagnostic tools**, and then **Logs**.

You should also refer to the Tivoli Identity Manager Version 5.1 Information Center found at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_admin_oview_requests.html

Click **Administrating** and then **Requests administration**.

Reports can also be run against the audit logs. Any action taken by a Tivoli Identity Manager user that changes a business object or the configuration of the system is audited:

- ▶ ACI management
- ▶ User management (people, access, role, and container management)
- ▶ Policy management (provisioning, service selection, identity, recertification)
- ▶ Service management
- ▶ Account management
- ▶ Configuration management
- ▶ Authentication events

Only non-workflow actions will be audited (workflow actions currently have an audit trail). Workflow actions are also audited, but only at the high level. The audit log for any activity can be viewed using the Tivoli Identity Manager Web user interface.

4.2.8 Reports

Tivoli Identity Manager provides 32 different standard reports, four of which are only available byway of Tivoli Common Reporting². These reports use predefined templates that enable you to specify criteria that produce the report details that you want:

- ▶ 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 request's 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.

² Go to the IBM Tivoli Identity Manager Information Center Version 5.1at http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/cpt/cpt_ic_reportspack_intro.html and click Configuring and administering IBM Tivoli Common Reporting.

- ▶ 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.
- ▶ Account report: A report that lists services that the user can select from to generate the 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 that is a member of the 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.
- ▶ 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 will contain 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.
- ▶ Access control information (ACI): 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 must 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 targets 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.
- ▶ Separation of duty policy violation report: A report that lists various separation of duty policy violations based on policy name and business unit rules.³
- ▶ Separation of duty policy definition report: A report that lists various separation of duty policies. Allows filtering based on policy name and business unit.³
- ▶ User recertification history report: A report that provides a history of user recertification. Allows filtering based on date range, start and end date, policy name, business unit, user, and recertifier.³
- ▶ User recertification policy definition report: A report that lists user recertification policies. Allows filtering based on policy name and business unit.³

Access to any of the Tivoli Identity Manager reports is defined by report ACIs. These ACIs govern the availability of reports for all users, including access permission to view reports and access permission to run reports. Access to reports within Tivoli Common Reporting is governed by the authorizations defined against the report set.

³ Tivoli Common Reporting only

4.2.9 Entity relationships

As a summary, the key Tivoli Identity Manager entities and their relationships are shown in the management model in Figure 4-7.

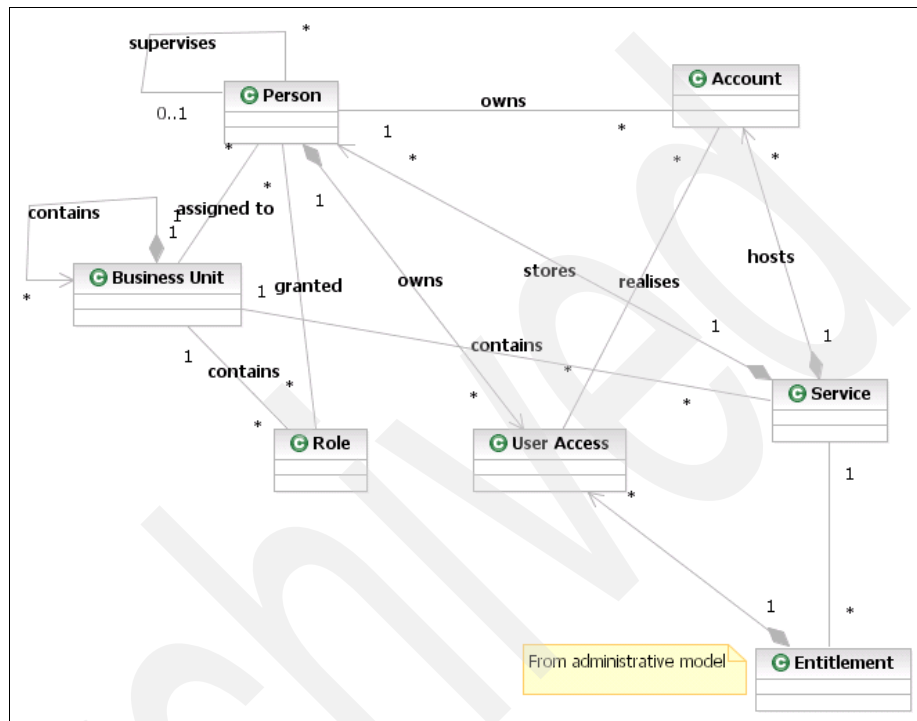


Figure 4-7 Tivoli Identity Manager management model

Person objects (also known as users) are supervised by other person objects. The supervisor relationship in Tivoli Identity Manager is a defined relationship used in a number of processes. For example, if a person does not have an e-mail address specified in his record but has a defined supervisor, his supervisor would receive notifications intended for that person. Additionally, a person may be granted permissions over the people who he supervises.

A person is defined in a *business unit*, which represents any identity Manager organizational tree element (such as organizational units or business partner units). They are associated with *roles*, which are granted to them, and *accounts* and user *accesses*, which they own.

Accounts are related to a *service*, which hosts them, and to user accesses, which are realized through them.

Provisioning policy entitlements are also dependent on services, of which they define the accesses for account provisioning. The administrative model for services and related objects is shown in Figure 4-8.

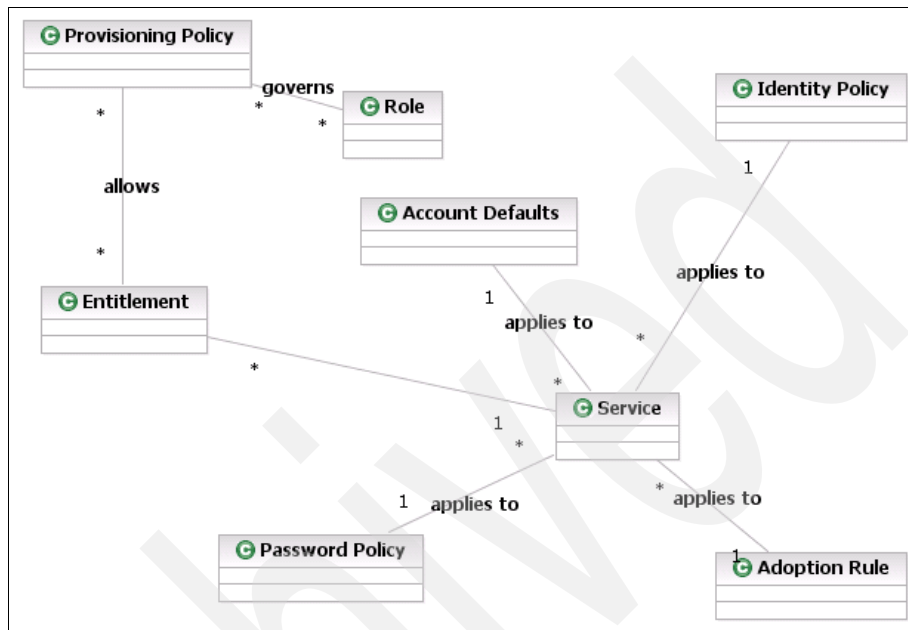


Figure 4-8 Tivoli Identity Manager administrative model

A provisioning policy allows an account entitlement to be defined to specify which account attribute values are allowed and disallowed when provisioning accounts. These policies are applied to people who are members of the organizational roles that they govern. Account entitlements are dependent on the service that they apply to, since services define account classes and attributes for the target resource that they manage.

Other objects that apply to services are:

- ▶ *Password policies*, which define the rules for account password generation for a service
- ▶ *Adoption rules*, which define how newly discovered accounts can be associated with Tivoli Identity Manager users
- ▶ *Account defaults*, which set attribute value defaults for a given set of services
- ▶ *Identity policies*, which define how account user identifiers should be created for a given set of service

4.2.10 Life cycle management

There are some processes that administrators must perform in order to manage the overall life cycle of accounts. For example, using the reporting utilities, you can create a report that lists all dormant accounts. But in order to delete an account, you first must consider additional information, for example, how long the account has been without activity, who is the owner of this account, whether this person still need the account to perform his assigned tasks, and so on. The goal of life cycle management is to provide rules and mechanisms to enable the automatic processing of this and similar events.

Life cycle rules

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 allow administrators to define events that will be triggered immediately or based on time intervals. The rules may also have matching criteria evaluated against an entity or entity type to reduce the scope of the target entities that the life cycle rules should be performed against.

For example, a life cycle rule could be created to check once a day for password owners who have not had their password changed in 90 days. An e-mail notification could be sent to the relevant people (that meet the life cycle rule search criteria for person criteria mentioned), informing them that they need to change their passwords.

Recertification process

In all environments, there are critical systems. Very often there are requirements that people with access to these systems must be recertified in order to retain their full access rights.

Recertification policies can help automate this specific process and allow recertification operations to be audited and reported on. A recertification policy could, for example, check all AIX accounts that must be recertified using a predefined interval, or on a rolling calendar basis. Account owners receive a notification, which would direct them to log into Tivoli Identity Manager. Once logged on, they can access an actionable item asking them to recertify their account.

The process above is a simple recertification policy that can be implemented using the base policy wizard. This recertification policy could be further refined by, for example, adding a second-level approval by the account owner's manager or the system owner. This would require the use of the advanced workflow design interface to customize the recertification workflow and implement the additional functionality.

Recertification policies can also be specified for the users and the person type, for example, *person* or *BPPerson*. This enables the recertification policy to recertify the persons' accounts, roles, and group memberships in one transaction. Figure 4-9 shows the resource targets selection window.

✓ General	Manage Policies > Manage Recertification Policies > Resource Targets
✓ Target Type	Specify the resources to which the recertification policy applies, and then click Next.
✓ User Target	
∞ Resource Targets	Recertify membership for the following roles
Additional Steps...	<input checked="" type="radio"/> All
	<input type="radio"/> None
	<input type="radio"/> Specified roles
	Recertify the following accounts
	<input checked="" type="radio"/> All
	<input type="radio"/> None
	<input type="radio"/> Accounts on specified services
	Recertify the following groups
	<input checked="" type="radio"/> All
	<input type="radio"/> All groups on specified accounts
	<input type="radio"/> None
	<input type="radio"/> Specified groups

Figure 4-9 Recertification policies: resource target selection

4.3 Tivoli Identity Manager functions

The key Tivoli Identity Manager functions are:

- ▶ Tivoli Identity Manager configuration and user self-care interfaces
- ▶ Password management
- ▶ Managing people and accounts
- ▶ Applying policies to people and account management
- ▶ Reconciling accounts
- ▶ Applying workflow to people and account management
- ▶ Setting up and apply workflow to access management

- ▶ Configuring Tivoli Identity Manager groups and views
- ▶ Producing reports
- ▶ E-mailing notifications
- ▶ Managing to-do lists
- ▶ Importing/exporting
- ▶ Managing groups

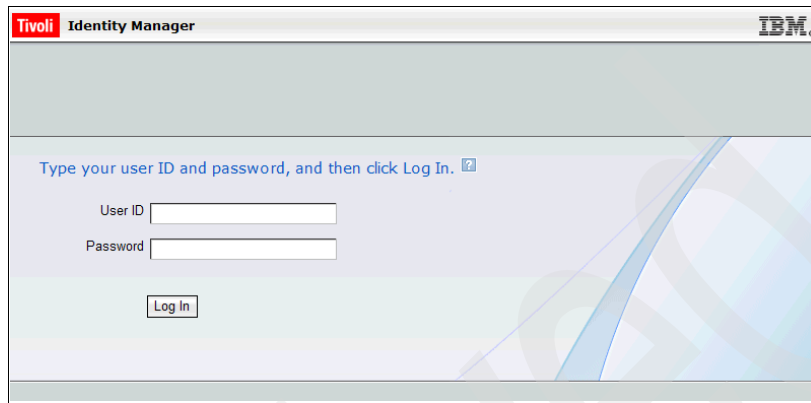
Tivoli Identity Manager provides two interfaces, one designed for Tivoli Identity Manager configuration and one for user self-care. All of the functions listed above are available in the Tivoli Identity Manager configuration interface. A subset of these functions is available in the Tivoli Identity Manager self-care interface. These functions are detailed in the following sections.

4.3.1 Tivoli Identity Manager configuration and user self-care interfaces

One of the key benefits of Tivoli Identity Manager allows users to maintain their account passwords and other personal information. Another key benefit is its flexibility and numerous configuration options. In recognition of this, Tivoli Identity Manager provides two interfaces, one dedicated to managing and configuring Tivoli Identity Manager, providing configurable accessibility to all of its features, and one dedicated to simple, user-friendly access to self-care features. Both of these interfaces are Web-based, with no need for client software (other than the standard supported Web browsers, such as Microsoft Internet Explorer or Firefox).

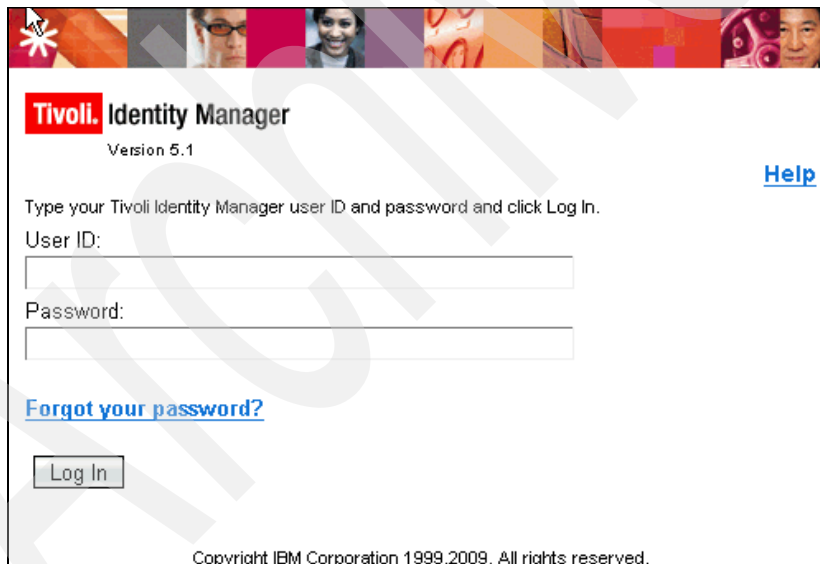
Login functions

The administrative console login page is shown in Figure 4-10. The self-care interface login page is shown in Figure 4-11.



The screenshot shows the administrative console login page for Tivoli Identity Manager. At the top left, it says "Tivoli Identity Manager" and at the top right, the "IBM." logo. Below the header, there is a light blue background with the instruction "Type your user ID and password, and then click Log In." followed by a small icon. There are two input fields: "User ID" and "Password". Below these fields is a "Log In" button.

Figure 4-10 Tivoli Identity Manager administrative console login page



The screenshot shows the self-care interface login page for Tivoli Identity Manager. At the top, there is a banner with several small images of people's faces. Below the banner, it says "Tivoli Identity Manager" and "Version 5.1". On the right side, there is a "Help" link. The main text says "Type your Tivoli Identity Manager user ID and password and click Log In." There are two input fields: "User ID:" and "Password:". Below these fields is a "Log In" button. There is also a link for "Forgot your password?". At the bottom, there is a copyright notice: "Copyright IBM Corporation 1999,2009. All rights reserved."

Figure 4-11 Tivoli Identity Manager self-care interface login page

Both feature an optional Forgot your password? option that uses the challenge/response function to verify the user before prompting for a password reset (the *forgot your password* functionality has been disabled in Figure 4-10 and enabled in Figure 4-11).

Self-service functions

The self-service section of Tivoli Identity Manager allows users to view and edit information that directly applies to them. Any person who is granted access to their own information can use the self-care interface to manage personal information and action items. The self-care interface is shown in Figure 4-12.

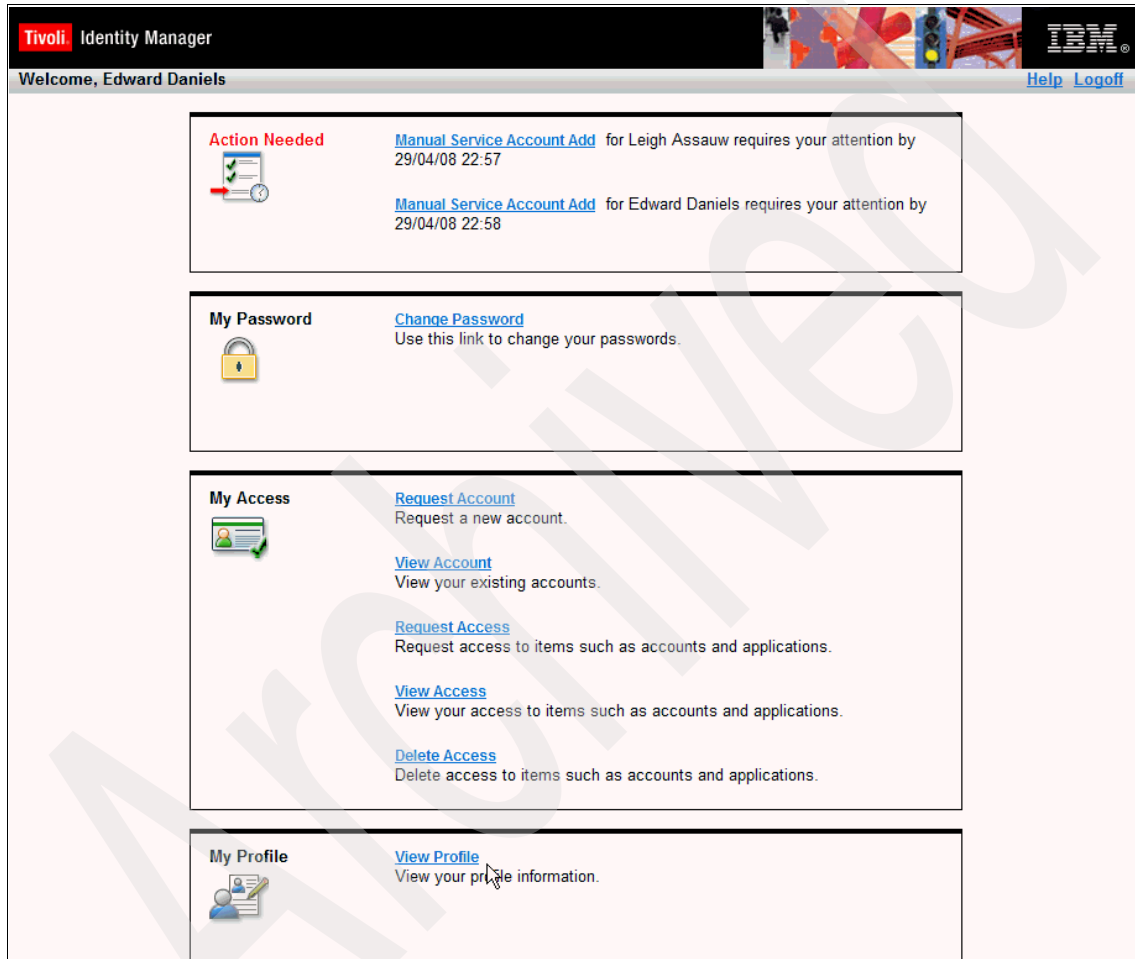


Figure 4-12 Self-care interface home page of a user

The self-care functions are:

- ▶ **My Password:** Users can manage passwords for one, multiple, or all accounts associated, as well as forgotten password challenge/response information if configured.
- ▶ **My Access:** Users can request new accounts and accesses, and they can manage their existing ones.
- ▶ **My Profile:** Users can manage their personal details.
- ▶ **My Requests:** Users can manage pending and completed requests.
- ▶ **My Activities:** Users can configure delegation of authority and see pending action items from workflows. These may be activities such as approvals, work orders, and requests for information. Note that active pending action items are also shown at the top of the interface in an Action Needed section.

Access controls can be used to restrict which of these functions can be viewed and accessed by each user.

These various functionalities are also available in the Tivoli Identity Manager administrative console, both for the user himself and for other users.

4.3.2 Password management

A key benefit to users is the ability to manage their passwords and related activities. Through Tivoli Identity Manager, users and administrators can centrally manage and synchronize their passwords across all their accounts.

Password synchronization

Tivoli Identity Manager provides flexibility around the management of password synchronization. Users can have the option to select the accounts on which the passwords are to be synchronized during a password reset, or to not be given the option and have all passwords synchronized during a password reset.

Reverse password synchronization

Tivoli Identity Manager provides integration with the native password reset functionality on various platforms. This is known as reverse password synchronization and allows the initiation of a password change to originate at the managed resource and can leverage Tivoli Identity Manager to verify that the new password meets the applicable password policy and apply the changed password to other accounts associated with the user. This integration is most commonly used in a Windows *Active Directory* environment. For example, when users reset their Windows passwords through their workstation, Tivoli Identity Manager can intercept this password change to ensure that it conforms to the password policy specified and, once verified, synchronize the user's password

with their other accounts. It is also available on various other platforms. Refer to the latest product documentation for a full list.

Challenge/response

If users forget their Tivoli Identity Manager account password, they can use the challenge/response feature to access their Tivoli Identity Manager account and reset their password. These are a set of questions defined within Tivoli Identity Manager specifically for situations where Tivoli Identity Manager account passwords are forgotten. Challenge/response questions can be defined by an Tivoli Identity Manager administrator (administrator defined) or personally by each user (user defined). When defined by an administrator, the set of challenge/response questions apply to all users with Tivoli Identity Manager accounts.

If enabled, the challenge/response function is initialized for users when they first log in to Tivoli Identity Manager through the administrative console, or is available as an action item for users logging into the self-care interface. If administrator defined, the users are prompted to select a number of predefined challenge questions from a list, as shown in Figure 4-13. These challenges can be configured to apply to a specific language or locale. They are defined by an Tivoli Identity Manager administrator in the administrative console.

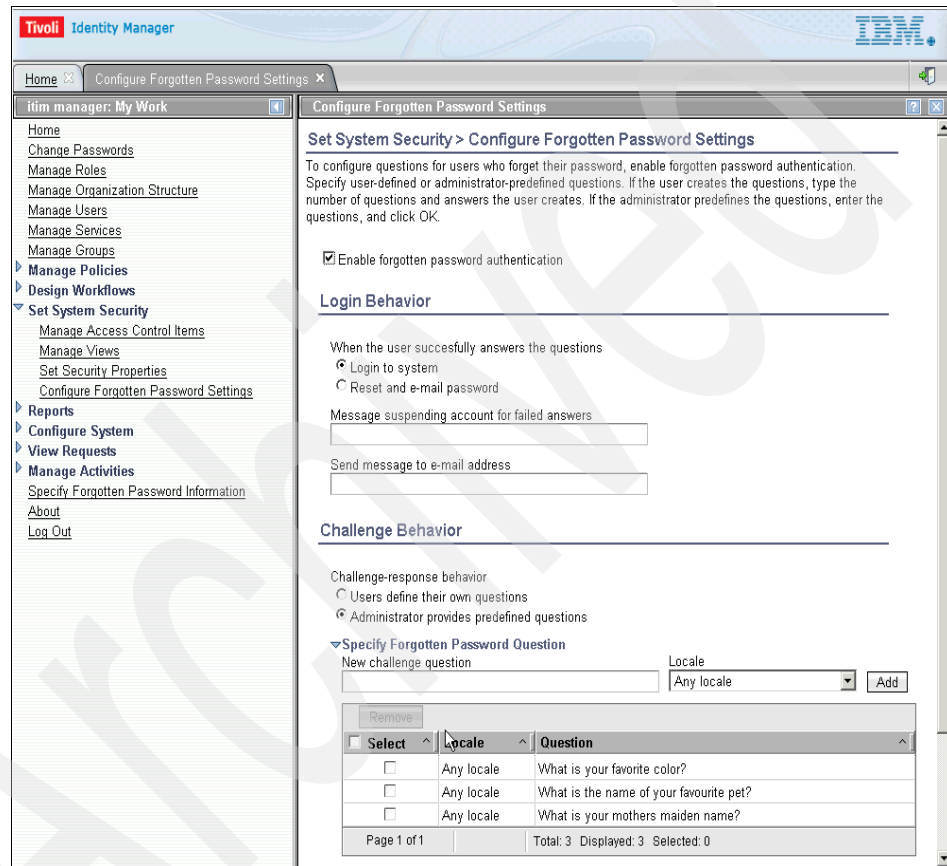
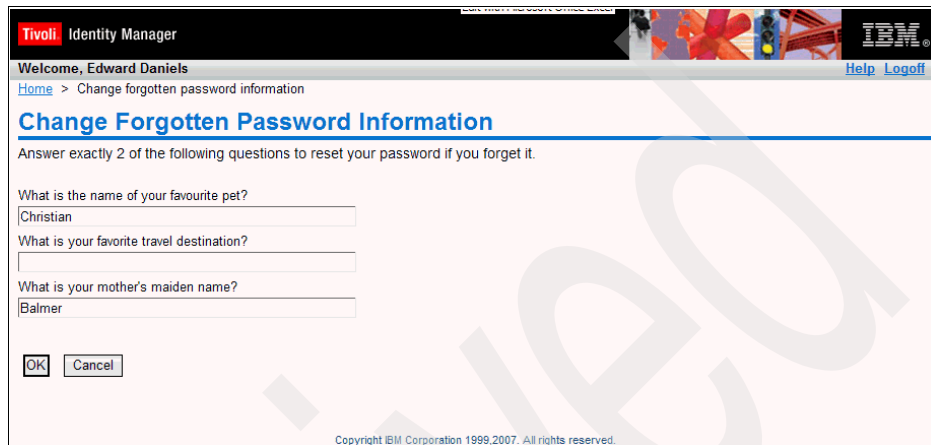


Figure 4-13 Challenge/response specification page

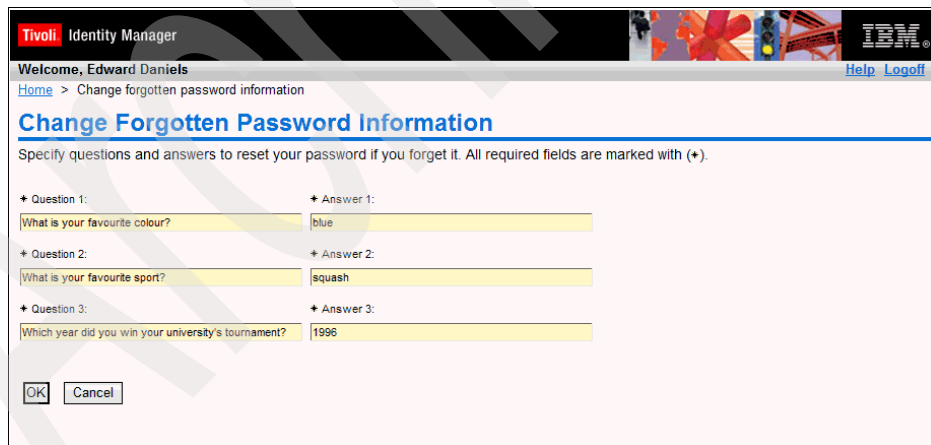
It is possible for the user to select the challenge/response questions and enter the response to each challenge question in both Tivoli Identity Manager interfaces. A user answering the challenges in the self-care interface is depicted in Figure 4-14.



The screenshot shows the Tivoli Identity Manager self-care interface. The header includes the Tivoli logo, the text "Identity Manager", and the IBM logo. Below the header, it says "Welcome, Edward Daniels" and "Home > Change forgotten password information". The main heading is "Change Forgotten Password Information". Below this, it says "Answer exactly 2 of the following questions to reset your password if you forget it." There are three questions with input fields: "What is the name of your favourite pet?" with the answer "Christian", "What is your favorite travel destination?" with an empty field, and "What is your mother's maiden name?" with the answer "Balmer". At the bottom, there are "OK" and "Cancel" buttons. A copyright notice "Copyright IBM Corporation 1999,2007. All rights reserved." is at the very bottom.

Figure 4-14 User's challenge/response details in the self-care interface

If user defined, they are prompted to provide their specialized set of challenge questions and responses, as shown in Figure 4-15.



The screenshot shows the Tivoli Identity Manager self-care interface with user-defined challenges. The header and navigation are the same as in Figure 4-14. The main heading is "Change Forgotten Password Information". Below this, it says "Specify questions and answers to reset your password if you forget it. All required fields are marked with (+)." There are three questions and answers, each marked with a red asterisk: "Question 1: What is your favourite colour? Answer 1: blue", "Question 2: What is your favourite sport? Answer 2: squash", and "Question 3: Which year did you win your university's tournament? Answer 3: 1996". At the bottom, there are "OK" and "Cancel" buttons.

Figure 4-15 User-defined challenges in the self-care interface

If they subsequently forget their password, they can select the **Forgot your password?** link on the login page of either the administrative console or the self-care interface. They will then be prompted to supply their responses before being asked to change their password, as shown in Figure 4-16.

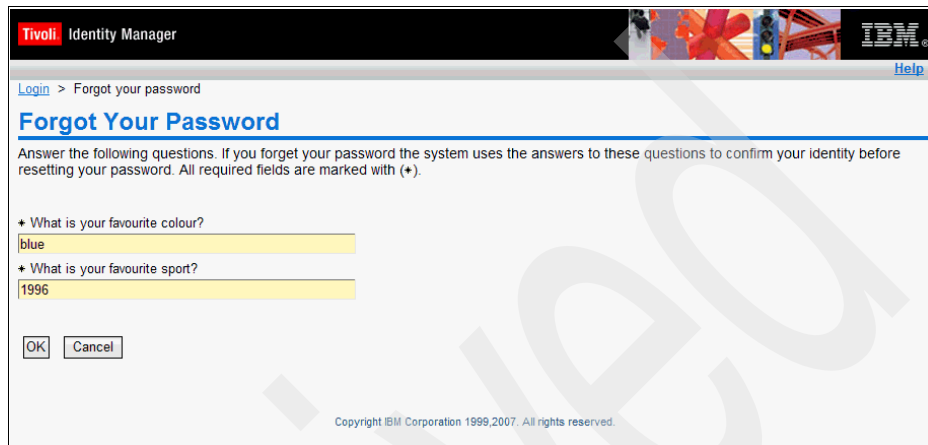


Figure 4-16 *Forgot password challenge responses in the self-care interface*

4.3.3 Manage people and accounts

Tivoli Identity Manager gives system administrators the ability to manage an organization's employees (people or users) from a central location. The "Manage Users" page is available through the side navigation bar.

From the Manage Users page, system administrators can perform the following tasks:

- ▶ Add a person.
- ▶ Modify a person.
- ▶ Delete a person.
- ▶ Suspend (deactivate) a person.
- ▶ Restore (activate) a person.
- ▶ Transfer a person (to a different container).

Manage people

Figure 4-17 on page 139 shows the Manage Users → Select a User tab.

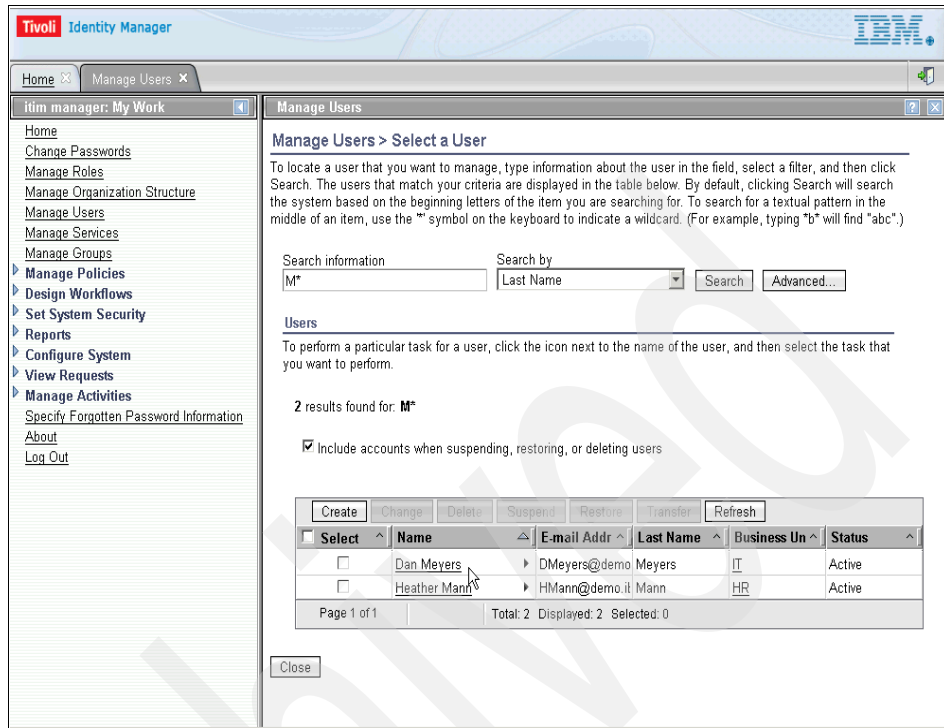


Figure 4-17 Manage Users page

The Manage Users page displays the names of the people in the selected branch of the organization tree, their e-mail addresses, their last names, their current organizational tree containers, and their current status. A search function is available at the top of the page.

Clicking the name of a person displays that person's profile, with access to its contents restricted according to the viewer's permissions. Clicking the arrow in front of a name displays a context-sensitive menu, which can be restricted as well. Menu options typically include modifying the corresponding person's record, request managing that person's account and accesses, and performing an operation on the person.

Account management

The Account Management page is used to manage accounts for a person. An account is a person's access to Tivoli Identity Manager or to a service (managed resource), such as Windows Active Directory, Solaris, and so on. There are multiple ways to manage accounts:

- ▶ Select a specific person for whom to manage accounts.
- ▶ Select a service instance for which to manage accounts.

Figure 4-18 shows the account management page for Dan Meyers from the Tivoli Identity Manager administration console.

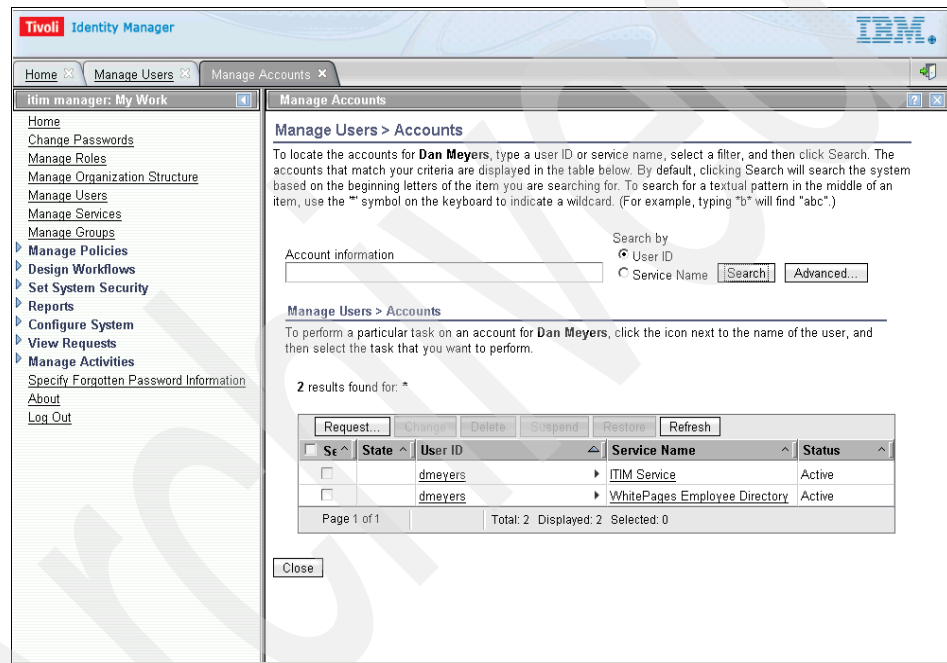


Figure 4-18 Person account management page in the administration console

A search function for accounts can be found at the top of the window to locate specific accounts.

Figure 4-19 on page 141 shows the account management page for Rob Hoffman in the Tivoli Identity Manager self-care interface.

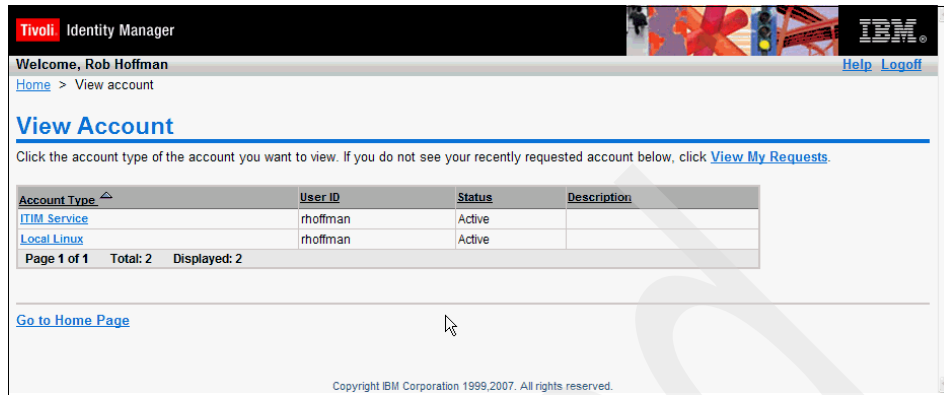


Figure 4-19 Person Account Management Page in the self-care interface

4.3.4 Apply policies to person and account management

Policies are used to determine and enforce compliance of people and their accounts managed by Tivoli Identity Manager. They are also used as the basis for automation of account provisioning and de-provisioning, account ID creation, and password strength checking.

The policies provided with Tivoli Identity Manager are:

- Provisioning policy** Dictates the accounts that can potentially be assigned to a person
- Identity policy** Defines user ID generation
- Password policy** Defines password rule and strength checking
- Service selection policy** Defines provisioning of accounts based on a person's attributes
- Adoption policy** Defines the association of newly discovered accounts on a managed resource with Tivoli Identity Manager person records
- Recertification policy** Defines the process used for certifying user accounts and accesses
- Separation of duty policy** Defines mutually exclusive relationships among roles

Using a provisioning policy for account provisioning

A *provisioning policy* is a mechanism used to control user account *entitlements*. It prescribes the types of accounts that people are allowed to own and be provisioned to. Each policy maps sets of people to a set of entitlements. The sets of people are segregated as follows:

- ▶ All people defined in Tivoli Identity Manager: The policy applies to everyone.
- ▶ People defined to particular organizational roles: The policy applies to people who are members of the specified roles.
- ▶ All people who are not a member of any other policies that grant them the same account.

The set of entitlements defines the accounts and respective attributes for each account type that is provisioned as part of the policy. An entitlement can contain logic that determines the values for the account attributes and pre-fills the information. This reduces the reliance on manual data entry for cases where accounts are manually provisioned. It also allows accounts created automatically to be defined using the attributes given in the entitlements.

This also means that accounts associated with a provisioning policy must continue to conform to the policy for the lifetime of the account until such a time when it is determined that a different policy must be applied to it. Accounts that do not conform to existing Tivoli Identity Manager provisioning policies are deemed to be non-compliant and are handled using the following methods:

Mark non-compliance	Flags all accounts that do not comply with existing provisioning policies.
Suspend non-compliance	Suspends all accounts that do not comply with existing provisioning policies.
Correct non-compliance	Updates the accounts as soon as their non-compliance has been determined, so that the attributes that do not comply with existing provisioning policies are updated to comply with them.
Alert non-compliance	Flags all accounts that do not comply with the existing provisioning policies and alerts a specified person to action the non-compliant account via their to-do list. They can accept the change, reject the change, defer the change to a later date, or do nothing about it.

Policy enforcement can be defined at a global level or at an Tivoli Identity Manager service level, which represents a single managed resource. Global level policy enforcement is applied by default to all services, but can be overridden by service-level policy enforcement.

The main implication of the role-based access control concept employed by the Tivoli Identity Manager provisioning policy model is that an individual person cannot be directly associated with a policy. People are typically assigned to a role that will then subject them to a policy evaluation based on the policies that apply to the role. The exceptions to this exist when the policy asserts that it applies to all Tivoli Identity Manager people or to all people who are not members of any other policies that grant them the same account.

Examples of how a provisioning policy can be used are:

- ▶ Where all people defined within Tivoli Identity Manager must have a specific account type, for example, the Tivoli Identity Manager account or a Windows domain account, an organizational-wide provisioning policy can be used.
- ▶ Where the set of accounts that a user gets is based on his job role or team, then a provisioning policy can be created to apply to people who are members of the relevant Tivoli Identity Manager organizational role. For example, there may be a provisioning policy tied to the *Windows system administrator* organizational role that provisions an administrative account on all Windows servers. There may also be a provisioning policy tied to the *HR administrator* organizational role that provisions accounts on the HR application system.
- ▶ You may also need a provisioning policy to cover users that are not covered by any other provisioning policy. For example, you have defined job-role-based provisioning policies for most of the organization, but you want other users to receive a basic set of accounts.

Automated provisioning of accounts is defined within a provisioning policy. There is an option to specify that account provisioning for a service must be manually executed using the Tivoli Identity Manager interface. It is important for a user to understand that with manual provisioning someone must request a new account in order to get the account. When automatic provisioning is used, it can be configured so that when a user's entitlements change (for example, through a job role change) the old accounts are automatically de-provisioned and new accounts are automatically provisioned (with common accounts maintained).

Figure 4-20, Figure 4-21 on page 145, and Figure 4-22 on page 145 show the provisioning policy definition for the default Tivoli Identity Manager service. This policy entitles all users in Tivoli Identity Manager to an Tivoli Identity Manager account.

The screenshot shows the 'Manage Provisioning Policies' page in the Tivoli Identity Manager web interface. The page title is 'Manage Policies > Manage Provisioning Policies > General'. The left sidebar contains tabs for 'General', 'Members', and 'Entitlements'. The main content area contains the following fields and options:

- Policy name:** Default provisioning policy for ITIM
- Caption:** ITIM account policy
- Make policy available to services in:** Radio buttons for 'This business unit and its subunits' (unselected) and 'This business unit only' (selected).
- Description:** Allow everyone to be provisioned for an ITIM account.
- Policy status:** Radio buttons for 'Enable' (selected) and 'Disable' (unselected).
- Priority (integer greater than 0):** 10000000
- Keywords:** (empty text field)
- Business unit:** Tivoli Austin Airlines

At the bottom of the page, there are buttons for 'Submit', 'Preview...', 'Save as Draft', and 'Cancel'. A search box is also visible next to the Business unit field.

Figure 4-20 Provisioning policy page: General tab

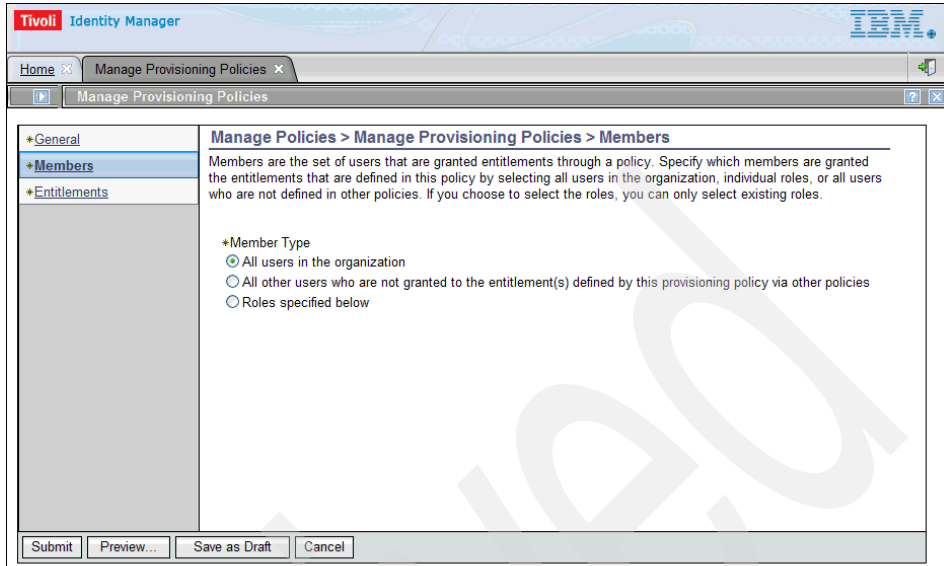


Figure 4-21 Provisioning policy page: Membership tab

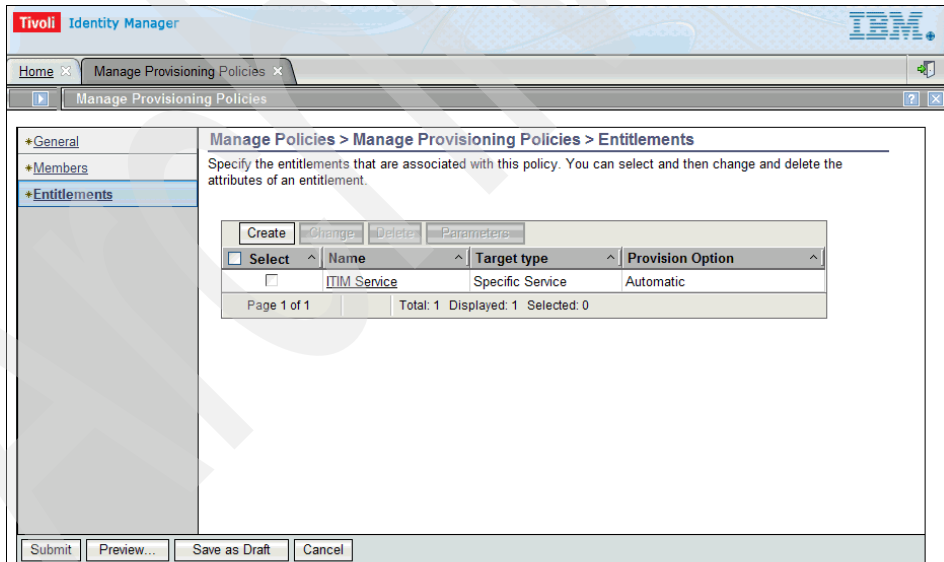


Figure 4-22 Provisioning policy page: Entitlements tab

The last tab shows the entitlements for the Tivoli Identity Manager service (for Tivoli Identity Manager accounts). In this example provisioning policy, Tivoli Identity Manager accounts are created automatically (the provision option is set to automatic), and there is no workflow associated with this policy.

There can be multiple services associated with each provisioning policy. For example, a policy for provisioning all Linux servers (for Linux system administrators) could contain a service for every Linux system. A user may be subject to multiple provisioning policies, depending on the scope of the policies and where the user is placed in the org tree.

A provisioning policy may be saved as a draft within the system. This allows the definition of provisioning policies to be implemented in multiple phases rather than making the user complete a provisioning policy definition in the one sitting.

There also exists the ability to preview the effect of a provisioning policy on users before adding, modifying, or deleting the policy, as shown in Figure 4-23 on page 147. Previewing a provisioning policy provides a summary of the number of accounts effected as well as specific details for each account impacted by the policy. Details of which accounts will be provisioned, suspended, deleted, modified, marked as non-compliant, or have their status changed from non-compliant to compliant are provided in the provisioning policy preview. If the results of the preview are as expected, the policy can then be submitted and activated. Conversely, if the results of the preview are not as expected, then it can be revised.

Manage Policies > Manage Provisioning Policies > Preview Policy Summary

Use this summary page to preview the impact of the provisioning policy on user accounts. You can click on the account links to view the details of the account changes. To stop evaluating the policy, click on Stop Evaluation button. This summary is automatically updated every 10 seconds until policy evaluation is completed or stopped.

Evaluation status: Completed
 Accounts evaluated: 6
 Error account: 0 accounts
 Provision new account: 0 accounts

▼ Disallowed account: 0 accounts

Enforcement Action	Number of Accounts
Alert account	0
Delete managed account	0
Mark account	0
Suspend managed account	0

▼ Noncompliant account: 3 accounts

Enforcement Action	Number of Accounts
Alert account	0
Mark account	3
Revoke privileges from managed account	0
Suspend managed account	0

▼ Compliant account: 0 accounts

Enforcement Action	Number of Accounts
Change managed account	0
Clear compliance flag	0

Stop Evaluation Close

Figure 4-23 Policy preview

Using an identity policy for account ID generation

An *identity policy* dictates how account logins or user IDs are generated. The supplied default identity policy checks whether a person has been assigned a preferred user ID, and if not, uses the first letter of the user's first name and the surname to generate an account ID. For example, the account ID for John Smith would be jsmith if he had no existing preferred user ID and there were no other jsmith's defined.

Identity policies can be set as global (applied to all services) or be set against specific service instances. The services an identity policy applies to are scoped by the organizational unit in which the identity policy is defined.

Identity policies can be defined dynamically either using a simple wizard or through the use of JavaScript for more complex policies. The JavaScript can make use of all standard JavaScript functions and programming constructs like loops and conditional branches. The values of personal attributes can also be retrieved with the Tivoli Identity Manager specific JavaScript functions. There is a supplied function to check whether the generated account ID is already in use in Tivoli Identity Manager defined accounts. Refer to “Identity policy script example” in the IBM Tivoli Identity Manager Information Center Version 5.1, for more details, at:

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

Using a password policy for password strength enforcement

A *password policy* is used for enforcing password strength. This applies to any password changes made through Tivoli Identity Manager. This includes password changes made at managed endpoint systems that are captured by the password interceptor component installed and sent back to the Tivoli Identity Manager server. For example, the Tivoli Identity Manager Windows Active Directory password interceptor can be enabled to be subjected to the password rules defined in Tivoli Identity Manager.

Password policies can be set as global (applied to all services) or be set against service types or specific service instances. There is a standard set of password rules that can be used to define a password policy. These can be extended using the password rule API. Refer to “Application programming interfaces” in the Reference section of the *IBM Tivoli Identity Manager Information Center Version 5.1*, for more details.

Figure 4-24 on page 149 shows the standard password policy window with options.

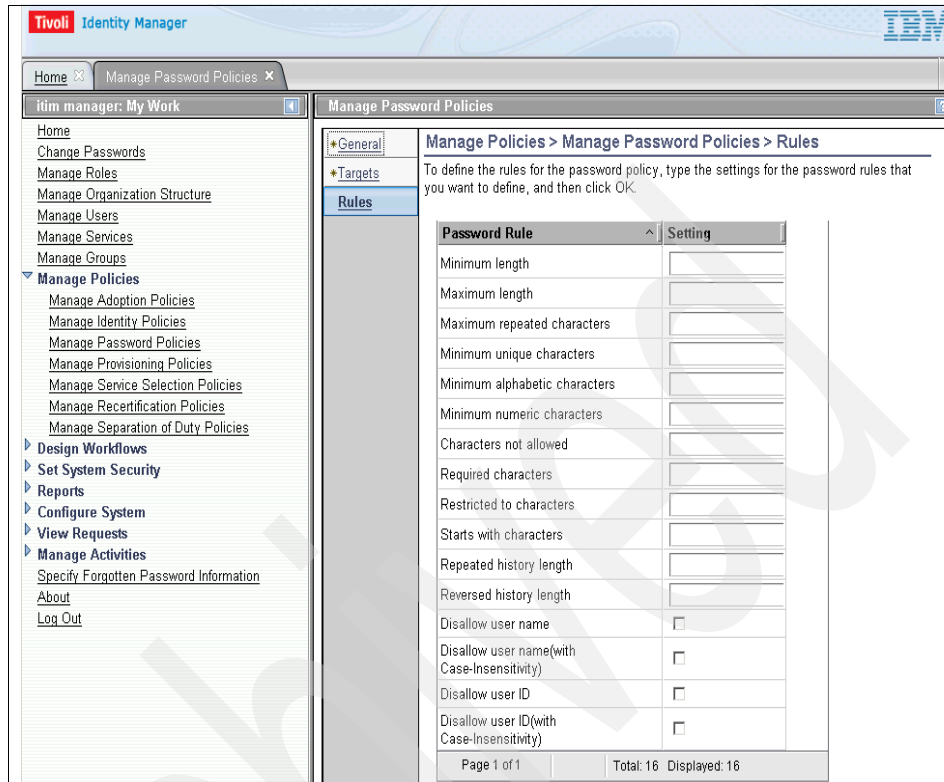


Figure 4-24 Password policy page: Rules tab

Using service selection policy for attribute-based provisioning

A *service selection policy* is defined using JavaScript. It is used when a user's attributes dictate on which particular service instance a person should be provisioned an account. For example, consider the case where all users in an organization have the same global provisioning policy that applies to grant them access to the Windows environment. The Windows domain that they are provisioned to depends on their department number. In this case, there is a Windows service instance defined for each Windows domain. The service selection policy is then used to determine the correct service instance to use for each user based on the user's department number, and thus provisioned to the relevant Windows domain for that department.

Using adoption policy to assign orphan accounts to users

An *adoption policy* is used to define how an account with no owner can be associated with a person in Tivoli Identity Manager. This is achieved by defining a logic that can retrieve information from the orphan account in question and use it to determine whether one and only one user's person record contains sufficient matching information. For example, a possible adoption policy takes an orphan account's user ID and attempts to match it to a person's preferred ID in his record, or if that does not return any results, to any alias values entered in his person record.

Adoption rules are used during reconciliation operations. Orphan accounts may be encountered during a reconciliation operation, either because it is a new account created since the last reconciliation operation, or because previous attempts to assign it an owner have failed. In any case, Tivoli Identity Manager invokes the corresponding adoption policy to attempt to match it with an Tivoli Identity Manager user. If no match is found, Tivoli Identity Manager keeps a record of the orphaned account, but will not manage the account, as it cannot associate a policy to manage it.

An account can also be adopted manually by having an Tivoli Identity Manager user with the appropriate permissions associate an orphan account with an Tivoli Identity Manager person. In the same way, a previously owned account can be manually orphaned.

Adoption policies can be configured at a global level and for specific services. Service-specific policies override global settings for these policies. Adoption policy logic can be entered through a wizard for simple field matches or using JavaScript for more complex logic.

Using recertification policy to validate accounts and accesses

A user's accounts and accesses (role and group memberships) may need to be periodically validated to ensure that their use matches a corresponding business need. A *recertification policy* is used to ensure that a process is in place to revalidate accounts and accesses assigned to users. For example, accounts on a sensitive platform, such as a payroll system, may need to be recertified by its owner every six months on a rolling calendar basis (rather than at fixed dates). If the owner chooses not to recertify the account, or does not decide to certify or recertify his account, it will be suspended. If he recertifies it, he will be granted access to the account for another six months, at the end of which the cycle starts over again.

The same scenario can apply to accesses granted to a person, for example, an administrator privilege on a specific platform. It may be granted on the condition that the account owner recertifies his accesses for this platform periodically.

Recertification policies can be created for accounts, accesses, and users. Figure 4-25 shows a sample recertification policy's target type tab, where you select the type of target.

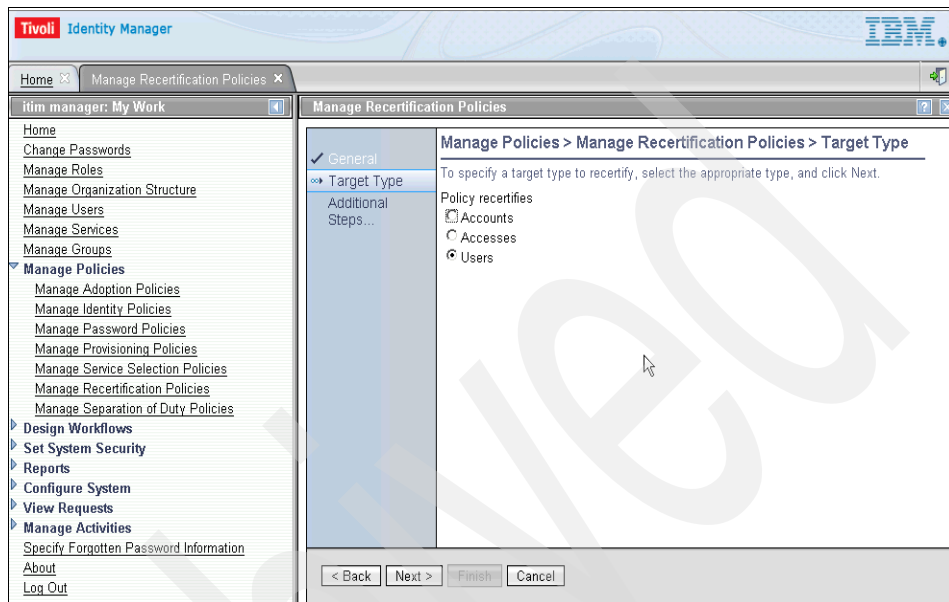


Figure 4-25 Recertification policy Target Type tab

Figure 4-26, “Recertification policy Service Target tab” on page 152 and Figure 4-27 show the policy’s next tabs, where you enter the service or user information. A single policy can apply to multiple services or accesses, but to only one user type (person, BPPerson or custom person) per policy.

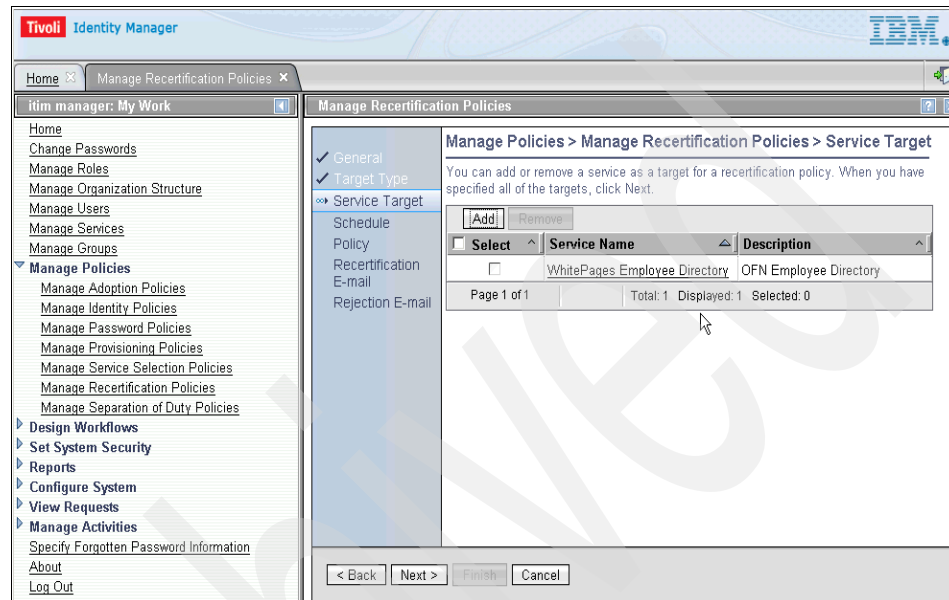


Figure 4-26 Recertification policy Service Target tab

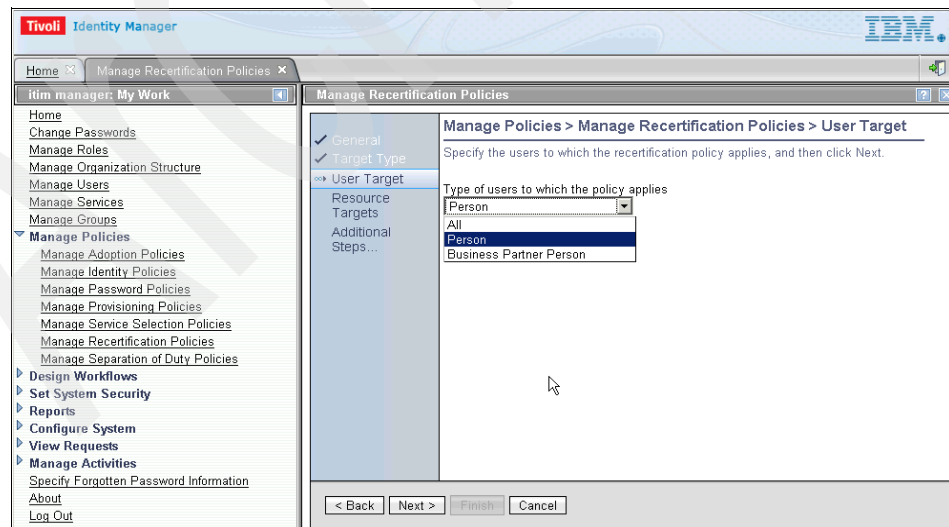


Figure 4-27 Recertification policy User Target tab

In the following tab, as shown in Figure 4-28, you define the recertification schedule as well as the policy check frequency. In our example, accounts must be recertified every 360 days, and the policy checks the configured target service's accounts daily at 00:00 (in the Tivoli Identity Manager server's time zone).

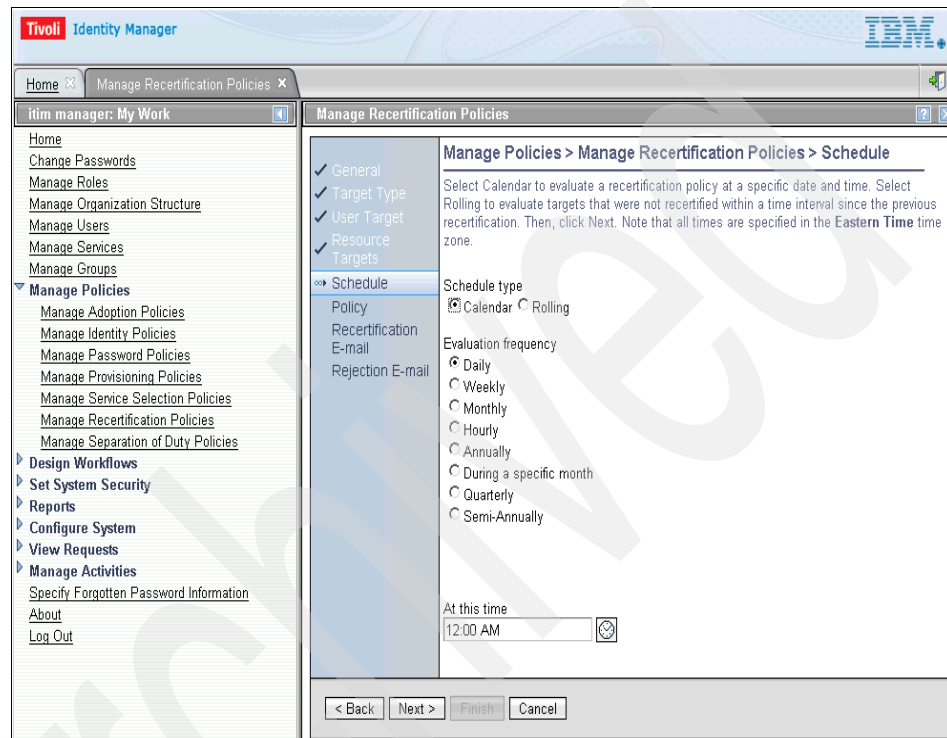


Figure 4-28 Recertification policy Schedule tab

The following tab, as shown in Figure 4-29, shows a policy wizard where you enter a simple workflow to define activities taking place for accounts in need of recertification. The tab's top option, simple/advanced, defines whether the workflow wizard or advanced workflow design interface will be used to define the policy's activities.

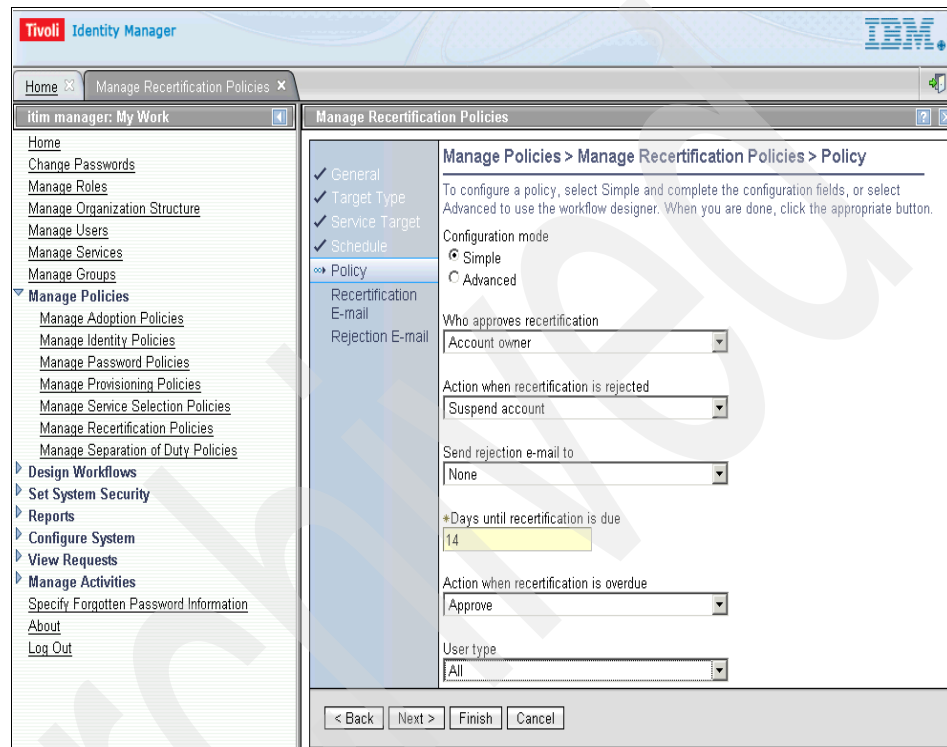


Figure 4-29 Recertification policy Policy tab

The remaining configuration items for the policy are the notification e-mails to be sent to recipients when an account is in need of recertification and when the account recertification has been rejected. E-mail templates can be selected or added to satisfy specific requirements.

Using separation of duty policy to detect conflict of interests

You may be required to define one or more *separation of duty policies* (SoD) to detect and manage conflicts of interest. Within these policies, you define two or more roles that, if owned by one person, would be a conflict of interest and trigger a policy violation.

Figure 4-30 shows the create separation of duty policy definition.

The screenshot shows the 'Create Separation of Duty Policy' form in the Tivoli Identity Manager interface. The left sidebar contains a navigation menu with options like 'Home', 'Manage Roles', 'Manage Organization Structure', 'Manage Users', 'Manage Services', 'Manage Groups', 'Manage Policies', 'Design Workflows', 'Set System Security', 'Reports', 'Configure System', 'View Requests', and 'Manage Activities'. The main content area is titled 'Manage Policies > Manage Separation of Duty Policies > Create Separation of Duty Policy'. It contains the following fields and sections:

- Policy name:** A text input field containing 'SoD for Online Banking System Administrators'.
- Description:** A text area containing 'Policy to stop system administrators being able to grant system administration role to others while also being in the system administrator themselves.'
- Business unit:** A dropdown menu showing 'Open Financial Network' and a 'Search...' button.
- Policy Rules:** A section with instructions: 'You can add, change, or delete policy rules. Select a policy rule in the table and then click the appropriate button.' Below this is a table with columns: 'Select', 'Description of Separation', 'Allowed Number of Roles', and 'Roles'. The table is currently empty, with a status bar showing 'Total: 0 Displayed: 0 Selected: 0'.
- Policy Owners:** A section with a checkbox 'Policy Owners' (checked), a 'Policy state' section with radio buttons for 'Enabled' (selected) and 'Disabled', and 'Submit' and 'Cancel' buttons.

Figure 4-30 Create Separation of Duty Policy

Assign a policy name and then add a description. The policy owner can be a person or an organizational role. You have to ensure that the policy owner is someone with the required authority within your business environment to make decisions on separation of duty violations. In some organizations this can be the security team.

Before saving the policy in either the enabled or disabled status, you need to create the policy rules by clicking the **Create** button. You may create one or more rules for the policy. Figure 4-31 shows the Create Policy Rule interface. Note the *allowed number of roles* can never be the same as the actual number of roles.

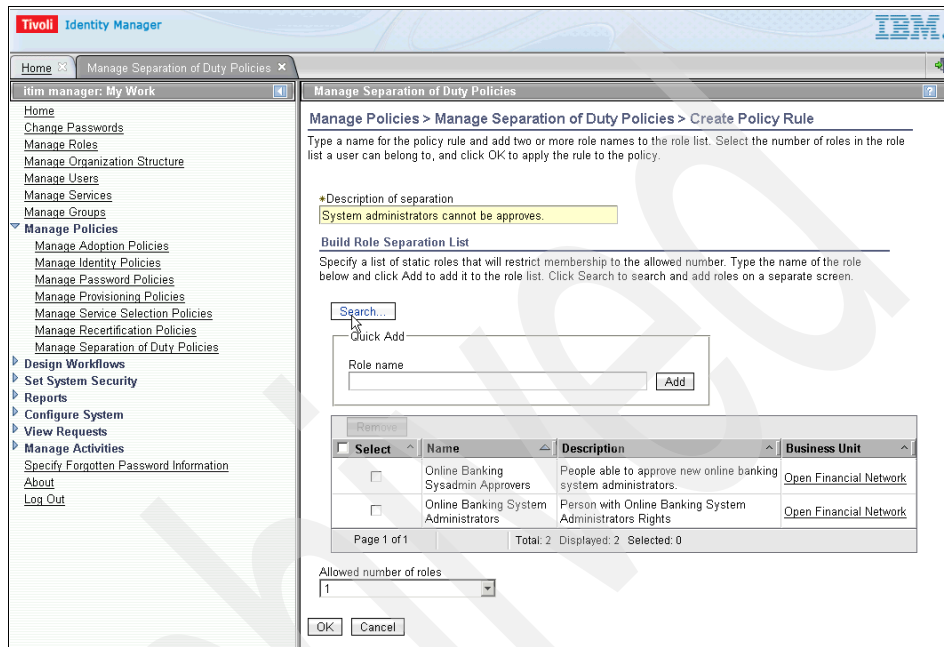


Figure 4-31 Manage Separation of Duty Policies: Create Policy Rule

Once a policy is created, you can manage it byway of the *Manage Separation of Duty Policies* menu option. After creating a policy, you should evaluate the policy. This will check whether there are any current policy violations that may require your attention. Figure 4-32 shows an example of the Manage Separation of Duty Policies interface. To evaluate your policy, search for the policy and then select it. Click **Evaluate** to start the evaluation process.

The screenshot displays the Tivoli Identity Manager web interface for managing separation of duty policies. The left sidebar contains a navigation menu with options like Home, Change Passwords, Manage Roles, and Manage Separation of Duty Policies. The main content area shows a search interface with a search information field and a search criteria dropdown (Name or description, Business unit, Role name). Below the search area, there is a table listing the search results. The table has columns for Select, Policy Name, Description, Business Unit, State, Violations, and Exemptions. One policy is listed: 'SoD for Online Banking System Administrators' with a description, business unit 'Open Financial Network', state 'Enabled', 0 violations, and 0 exemptions. The interface also includes buttons for Create, Change, Delete, Evaluate, and Refresh, and a Close button at the bottom.

Select	Policy Name	Description	Business Unit	State	Violations	Exemptions
<input type="checkbox"/>	SoD for Online Banking System Administrators	Policy to stop system administrators being able to grant system administration role to others while also being in the system administrator themselves.	Open Financial Network	Enabled	0	0

Figure 4-32 Manage Separation of Duty Policies

Any violations will be shown in the violations column (you may need to click **Refresh** to update the display). If violations are detected, you can access the violations summary form by clicking the violations count. From this window, you can view the details of the violations and exemptions. You may also approve or revoke exemptions if appropriate. See Figure 4-33.

The screenshot shows the Tivoli Identity Manager interface. The main content area is titled 'Manage Separation of Duty Policies > Violations and Exemptions Summary'. It states: 'The following are summaries of violations and exemptions by policy rule for the policy SoD for Online Banking System Administrators. Click on a specific policy rule name to see details about the violations and exemptions for that policy rule.'

Summary statistics:

- Total number of violations: 1
- Total number of exemptions: 0

Order rules: By violation (dropdown menu) [Sort]

System administrators cannot be approves. 1 Violations 0 Exemptions

1 Violations for Rule System administrators cannot be approves.

Select	Date of Violation	User Name	User Roles in Conflict	Policy Roles in Conflict
<input type="checkbox"/>	September 8, 2009 5:09:30 PM	Mike Stevens	Online Banking Sysadmin Approvers, Online Banking System Administrators	Online Banking Sysadmin Approvers, Online Banking System Administrators

Page 1 of 1 Total: 1 Displayed: 1 Selected: 0

0 Exemptions for Rule System administrators cannot be approves.

Approve section:

Select	User Name	Approve	Date Approv	User Roles in Confl	Policy Roles in Con	Approval Notes
Total: 0 Displayed: 0 Selected: 0						

Figure 4-33 Separation of duty, violations, and exemptions summary

4.3.5 Reconcile accounts

Reconciliation is the process of synchronizing the accounts and supporting data in Tivoli Identity Manager with the accounts and supporting data on a managed resource or endpoint. It is the Tivoli Identity Manager *discovery process* that queries the state of the accounts on the managed endpoint. To determine an owner relationship, reconciliation compares the account information with existing user data stored on the Tivoli Identity Manager server by first looking for the existing ownership within the Tivoli Identity Manager server and, secondly, applying adoption rules configured for the reconciliation.

Reconciliation is run for the following reasons:

- ▶ Load access information into Tivoli Identity Manager.

When a service is first integrated into Tivoli Identity Manager for management of the managed endpoint's accounts, there must be an initial load of 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 must be run to monitor the state of accounts and note whether 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 this means.

There are three main ways to run a reconciliation:

- ▶ Run a full reconciliation.

A full reconciliation reconciles all account and supporting data information. Supporting data information includes all account attributes for which Tivoli Identity Manager must keep a record to manage them, such as account groups, directory path, and available account profiles or types.

- ▶ Run a supporting data reconciliation.

A supporting data reconciliation reconciles only supporting data information. Supporting data information includes all account attributes for which Tivoli Identity Manager must keep a record to manage them, such as account groups, directory path, and available account profiles or types.

- ▶ Run a filtered reconciliation by defining a query.

A filtered reconciliation, also known as a query reconciliation, will only return accounts and supporting data information that matches the defined query filter. Only Tivoli Identity Manager account records satisfying the filter will be processed during the reconciliation. Other accounts will not be processed.

Reconciliations can be scheduled to occur during various intervals. Additionally, it is possible to only manage a subset of all account attributes during a reconciliation. By default, all account attributes are managed. Figure 4-34 represents a typical Tivoli Identity Manager service reconciliation window. Note that when the None option is selected in the query menu, none of the subsequent configuration options appear.

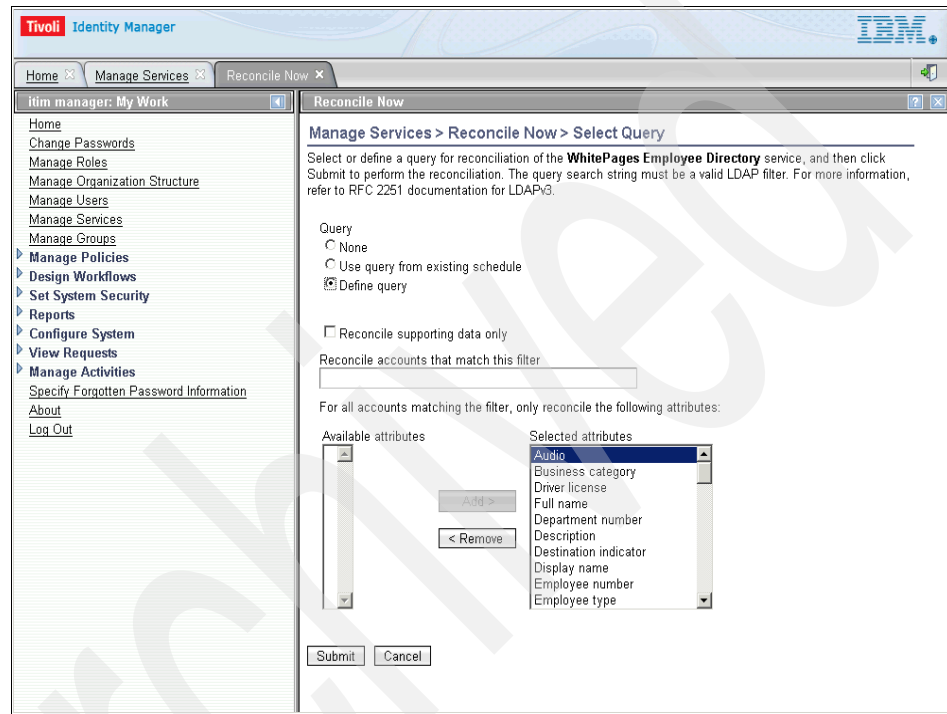


Figure 4-34 Reconciliation configuration window in the administrative console

During a reconciliation, if an entry for a managed account already exists within Tivoli Identity Manager, the reconciliation process updates the Tivoli Identity Manager entry to match the account on the managed resource. If an entry for a managed account does not exist, Tivoli Identity Manager creates one and attempts to associate it with an Tivoli Identity Manager person. If the association attempt is unsuccessful, the account is marked an *orphan* and will not be managed by Tivoli Identity Manager until it is associated with a person. If the association attempt is successful, Tivoli Identity Manager evaluates whether the account is compliant with the permissions and corresponding policies to which the user is entitled. If this is not the case, Tivoli Identity Manager acts according to the policy enforcement configuration as set for the managed resource's service.

For example, if a person's Windows account has been modified since the last reconciliation by a source external to Tivoli Identity Manager (for example, he has been added to the Windows HR group), this is then updated in the account record within the Tivoli Identity Manager directory accordingly. If a policy states that the person should not be a member of any groups within Windows, Tivoli Identity Manager can taking one of the following actions:

- ▶ 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.
- ▶ Correct the Windows account to no longer be a member of the human resources (HR) group within Windows.

4.3.6 Apply workflow to people and account management

A workflow is a set of steps or activities that define a business process. Tivoli Identity Manager workflows are a technical representation of specific business processes and can be used to complement account provisioning and life cycle management activities, such as adding, removing, and modifying people and accounts in Tivoli Identity Manager and managed resources across the environment. The use of Tivoli Identity Manager workflows centralizes the user and account management processes of an organization in addition to centralizing the auditing of these activities. This allows organizations to have a central point of control for user management and account provisioning and a central audit capture point to assist with meeting their audit and compliance requirements.

There are three main types of workflows in Tivoli Identity Manager:

- ▶ Account and access request workflows

Account and access request workflows are used to define how resources (accounts, accesses, and so on) are provisioned. Account request workflows are specified in provisioning policies, where they are associated with entitlements (hence their being referred to sometimes as entitlement workflows). Access request workflows are configured in individual accesses definitions.

Account and access request workflows add decision points to requests that add or modify an account or access. If the request is approved, the processing continues. If the request is rejected, the request is cancelled.

- ▶ **Operation and life cycle workflows**

Managed objects in Tivoli Identity Manager are called entities. Categories or classes of managed objects are called entity types, for example, accounts, persons, or business partner persons. 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 can be used to extend and create user-defined entity operations, which override those entity type operations.

Operation workflows are used to define the life cycle management of accounts and people. That is, operation workflows are used to add, delete, modify, restore, and suspend system entities, such as accounts and people. New operations that are required by business processes can also be added to the standard set mentioned. For example, a business process might require new accounts to be approved, so an operation workflow can be created that defines the set of activities that are needed to approve the account, including notifications and manager approvals.

Generally, operation workflows are used whenever the business process relates to the type of entity to be managed, such as accounts or people, or to the specific service type, such as all Linux systems.

Life cycle policies use some operation workflows to perform business processes periodically on a specified entity type or entity.

- ▶ **Recertification workflows**

Recertification workflows are defined in recertification policies. They define the recertification process of the policy. This recertification can apply to accounts, accesses or users, and usually consists of a recertification approval, followed by a recertification status being set in the account or access undergoing revalidation.

Unlike other types of workflows, recertification workflows are embedded directly in recertification policies.

Provisioning policy workflows

Another way to think of an entitlement workflow is to think of it as being associated with provisioning policies. A provisioning policy is tied in with sets of users that it applies to and the associated entitlements, that is, what accounts the users can have and exactly how these accounts should look. Entitlement workflows are directly related to this concept. Each entitlement can have an associated entitlement workflow that is executed for that entitlement. This implies that any provisioning policy can have multiple entitlement workflows associated with it.

User life cycle management workflows

User life cycle management relates to the activities surrounding a person and their accounts specifically relating to the following operations:

- ▶ Add
- ▶ Delete
- ▶ Modify
- ▶ Restore
- ▶ Suspend
- ▶ Self register (only relates to a person)
- ▶ Transfer (only relates to a person)
- ▶ Change password (only relates to a person's account)

Each of these operations relates to Tivoli Identity Manager entity types and has an associated operation workflow. There are four main Tivoli Identity Manager entity types, but we will limit the discussion to the following two:

- ▶ Person
- ▶ Account

Refer to “Configuring operational workflows with entities and entity types” in the IBM Tivoli Identity Manager Information Center Version 5.1, for more details on entity types, available at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_operations_oview.html

As indicated, certain operations are relevant to both entity types, and some are relevant to only one of the entity types. Each operation and entity type pair has an associated operation workflow. For example, there is a single operation workflow for the add operation on a person, and a different operation workflow for the add operation on an account. There may also be operational workflows associated with the specific entities associated with each entity type.

Life cycle policies

Operation workflows can also be associated with life cycle policies. These 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 re-occurring events such as password expiration, or contract expiration, which are driven by business policies. Life cycle policies can also eliminate the potential of some policies to go unenforced. Establishing life cycles allows 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 will be executed as a result of that event.

All life cycle policies consist of two parts, the definition of an event that triggers the rule and the identification of the life cycle operation that executes the actions specified in the rule. A life cycle policy may, for example, check periodically for persons whose contract have just expired, and suspend them.

Workflow interfaces

All workflows are configured in Tivoli Identity Manager's administration console. There are two interfaces available for create workflows:

- ▶ A workflow wizard allows a user to select a number of predefined activities from a list.
- ▶ A graphical workflow design interface allows workflows to be defined graphically.

Workflow wizard

A workflow wizard allows a user to select a number of predefined activities from a list. The list of activities is usually dependant of the type of workflow that is being created. For example, an account request workflow wizard and an account recertification wizard offer different options, each relevant to the policy or process to which they are associated. This workflow input option enables non-technical users to easily define simple workflows. Figure 4-35 shows an example.

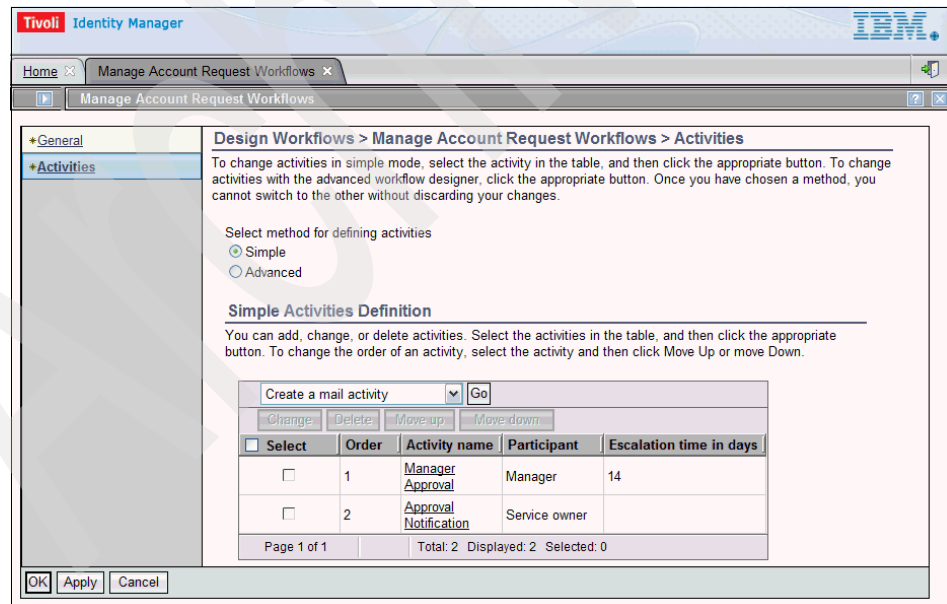


Figure 4-35 Simple account request workflow

Graphical workflow design interface

A graphical workflow design interface allows workflows to be defined graphically by dragging individual workflow elements from a menu and dropping them into a work area. These elements are then connected together to form a coherent chain of activities. This option enables users to define complex workflows with more flexibility than with the wizard interface, but requires more advanced technical skills, such as the need for some JavaScript knowledge. Figure 4-36 shows an example.

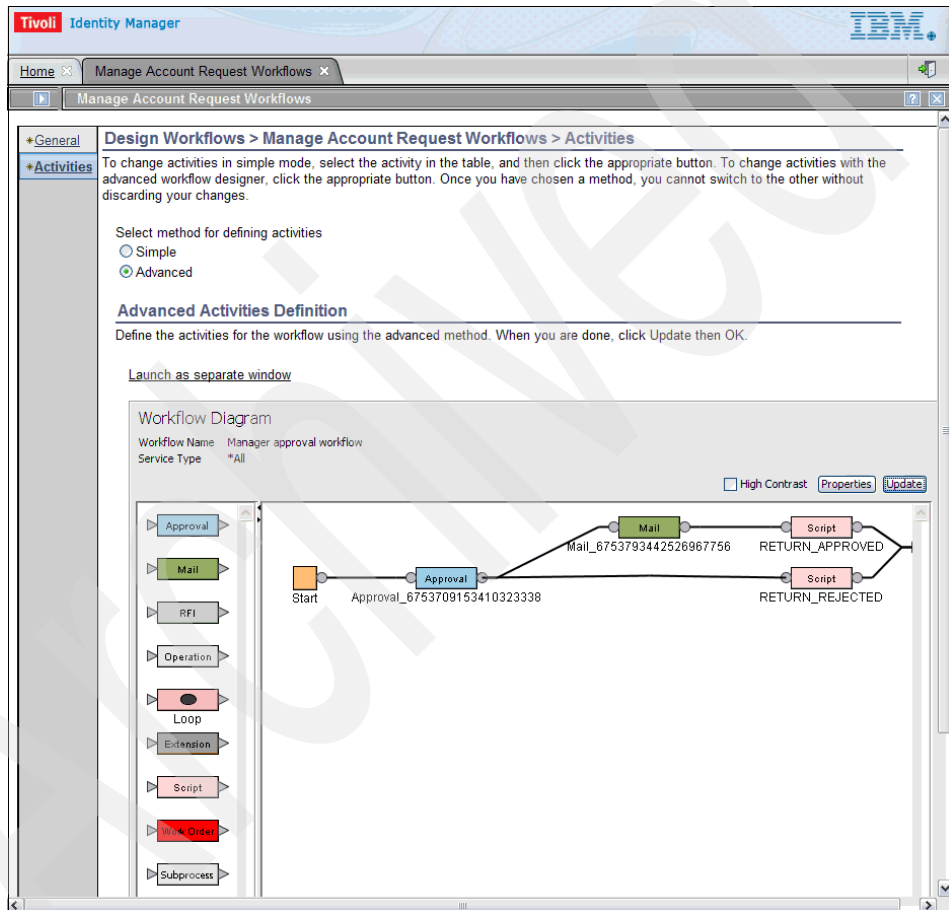


Figure 4-36 Advanced account request workflow

Graphical workflow elements

As with any high-level concept, a workflow can be defined in many ways, depending on the context. Any general concept of a workflow can be broken down into components that form its foundation. A workflow in Tivoli Identity Manager is no different. An Tivoli Identity Manager workflow can be thought of as consisting of the following high level-building blocks:

Processes	Definition of the activities, transitions, and data required to complete an end-to-end business process.
Activities	Business logic for a specific task in the workflow process.
Transitions	Flow between two activities driven by a condition. These flows can be defined to run in parallel or serially.
Input/output parameters	Data passed into and returned from a complete workflow process.
Relevant data	Global variable data for a workflow process.
Activity participants	Tivoli Identity Manager users that have been assigned to interact or action activities within a workflow process. This includes activities such as approvals, requests for information, and work orders.
JavaScript	Used for finer-grained control of elements within a workflow process.

Graphical workflow activities

Each element within a workflow plays a part. Some elements perform a more important role than others and as a result are more complex to define and require more detailed attention. In the case of Tivoli Identity Manager, the most important part of a workflow process is the business logic contained within it. Tivoli Identity Manager defines this using workflow activities. There are different types of workflow activities that can be used within a workflow:

- ▶ **Start and end nodes**

Start and end nodes are always included in a workflow and cannot be deleted. The start node defines the beginning of a workflow and the end node defines the end of a workflow.

► Approval

Used to add a request for approval for adding or modifying people and accounts. The approver must be an Tivoli Identity Manager user, as she must log in to Tivoli Identity Manager to approve or reject the request. An approval can also have an escalation participant. That is, if the primary approver does not action the approval within a specified amount of time, the approval will be escalated to the specified participant for her to approve. In entitlement workflows, approval nodes are typically used to request authorization to continue with a provisioning request. In operation workflows, approval nodes are typically used as a switch to follow a specific workflow path. Approval text and labels can also be modified to allow approvals to be used for most yes/no decision activities. For example, an approval node can be modified for use as a recertification activity. In the modified activity, the labels in the to-do list can be modified to indicate recertification and the choices can be changed from approve/reject to recertify/delete.

► Mail

Used to send a notification to an Tivoli Identity Manager. Mail nodes send the notification e-mail and moves on in the specified workflow activity.

► Request for information (RFI)

Used within entitlement and operation workflows to solicit account-related or person-related information from a person with an Tivoli Identity Manager account. The attributes that the participant is being asked to provide values for is specified within the definition of the RFI. Only the selected attributes will be editable by the participant. All other form attributes are read-only. The page displayed in the to-do list will match the form specified using the form designer. Attributes listed as mandatory on the account form will also be mandatory for the RFI. ACI definitions do not need to be created for the fields that the participant is being asked to respond to. An RFI can also have an escalation participant. That is, if the primary approver does not action the RFI within a specified amount of time, it will be escalated to the specified participant for them to action.

► Loop

Used to execute one or more nodes in a loop (nested loops are not supported). The activities defined within the loop will be executed repeatedly based on the loop condition. The condition defined in the loop specifies the activities to repeat while or until a specified condition is met. The condition defined in the loop must evaluate to true or false.

- ▶ Subprocess

Used to execute one entitlement workflow from another. Subprocesses simplify the workflow by using one node to represent a previously defined workflow sequence. Note that this is not available for use in operation workflows. Subprocess nodes are typically used for ease of organization (simplifies workflow layout) and reusability (common workflows can be leveraged in multiple workflows).
- ▶ Work order

Used to send a notification to an Tivoli Identity Manager user either as a notification or to request some type of manual activity. Work orders can be sent in two modes. The first mode is identical to the mail node functionality. The second sends the notification e-mail and places a to-do item in the user's to-do list. The workflow process does not continue until the user actions the work order to-do item or initiates an external process that uses the Tivoli Identity Manager API to complete the to-do item. An example of when this may be used is in the case where a building access badge must be created for a person. The audit trail will exist in Tivoli Identity Manager but the actual creation of the badge is done externally to Tivoli Identity Manager through a manual process.
- ▶ Script

Used to add logic to the workflow through the use of JavaScript code. It makes clear to anyone viewing the workflow that scripting is present in the workflow. JavaScript code is used within workflows to dynamically define and retrieve parameter and attribute values and to store and forward these values as variables for use by logic or code within a single workflow activity.
- ▶ Application extension

Used to invoke an application extension from within the workflow. An application extension is a Java class that has been pre-configured to be used in the workflow environment. Extensions can accept input parameters and return output parameters back to the workflow. Only extensions that have been properly registered appear in the extension window.
- ▶ Operation

Used to invoke an existing operation from within a workflow. The operation must be predefined for an entity type or entity. Operations take input parameters, but they do not return anything to the calling workflow.

Graphical workflow design and construction

There must be an initial requirements and analysis phase conducted in any Tivoli Identity Manager deployment before reaching the stage of specifying the relevant workflows within Tivoli Identity Manager. This is specific per organization, as the workflows are typically based on existing business processes that are directly mapped to Tivoli Identity Manager workflows or re-engineered for efficiency and subsequently defined in Tivoli Identity Manager.

Workflow construction is done through a simple drag-and-drop user interface. The workflow elements and activities are used to build workflows using this interface. An example of an entitlement workflow is shown in Figure 4-37. The interface for operation workflows is similar.

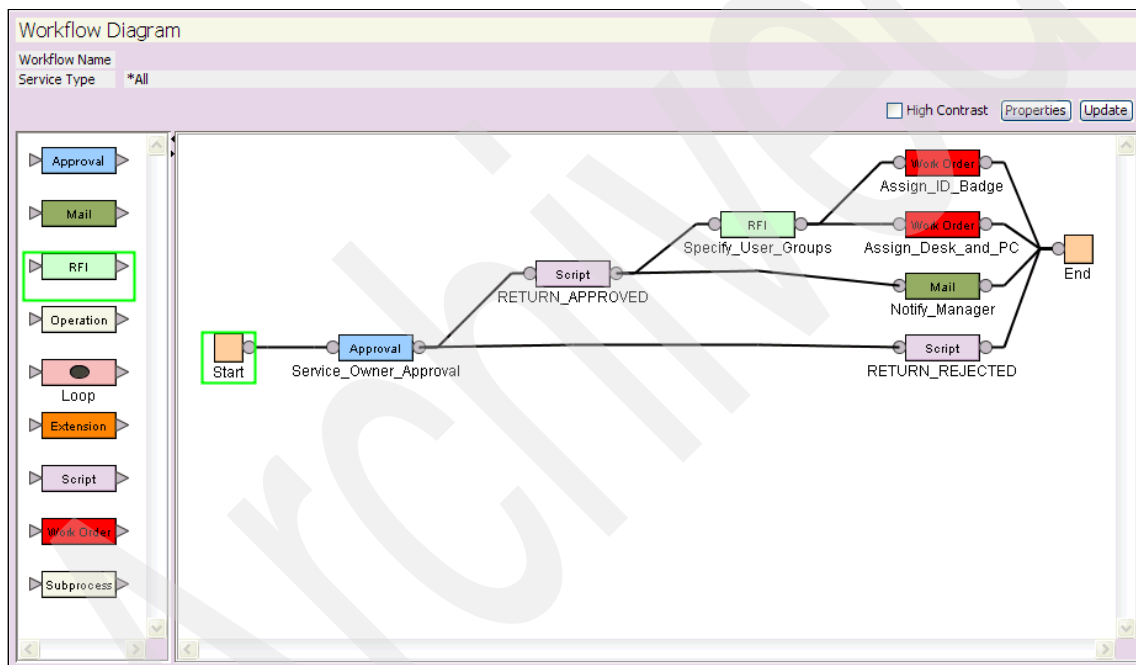


Figure 4-37 Advanced workflow interface

Let us step through Figure 4-37 as an example:

1. The *start* node is the initiation point of the workflow. Every workflow must have a start node.
2. In the *approval* node the manager of the person who the entitlement is being requested for (defined within Tivoli Identity Manager) must approve the request for the entitlement, for example, a Linux account.

3. If rejected, a final *script* node is invoked and the workflow ends with the person not receiving the entitlement. If approved, a *script* node is invoked and the workflow continues to two parallel tasks, a mail node and an RFI node. The mail node notifies the manager of the person who the entitlement is being requested for (defined within Tivoli Identity Manager) that an account has been created for him. The *RFI* node allows a specified participant to define the groups that the person is to receive within the entitlement being provisioned, for example, the Linux groups that the person is to become a member of upon provisioning.
4. Two further parallel manual tasks are then initiated from the RFI node, in the form of *work orders*—one to assign the person a desk and a workstation, and another to assign the person his building access badge. Once both are completed, the workflow completes with the person having been provisioned with his entitlements, building access, desk, and workstation. All the relevant steps are fully audited within Tivoli Identity Manager.

4.3.7 Produce reports

An authorized user can use the Tivoli Identity Manager reporting system to access and run reports to assist with the organizational auditing and compliance requirements. Tivoli Identity Manager contains a set of standard reports and also allows for the addition of custom reports. There is an interface provided to easily create a set of custom reports in cases where auditing requirements cannot be met by the standard set of reports. Both the standard and custom set of reports can be modified and deleted by an authorized user. The reports are generated in Portable Document Format (PDF) format or as comma-separated value (CSV) files, which can be used to feed into other reporting packages and their related data stores. Integration with Crystal Reports is provided and, if this is utilized, then the reports are generated in HTML format using the Crystal Reports Viewer. An integration with Business Intelligence and Reporting Tools⁴ (BIRT), an open source Eclipse-based reporting system, as well as Tivoli Common Reporting, is also possible.

The thirty-two standard reports are:

- ▶ Account operations
- ▶ Account operations performed by an individual
- ▶ Approvals and rejections
- ▶ Operation report
- ▶ Pending approvals
- ▶ Rejected report
- ▶ User report
- ▶ Account report

⁴ To learn more about BIRT check go to following link: <http://www.eclipse.org/birt/phoenix/>

- ▶ Accounts/access pending recertification report
- ▶ Individual access
- ▶ Individual accounts
- ▶ Individual accounts by role associated with provisioning policy
- ▶ Recertification change history report
- ▶ Suspended individuals
- ▶ Reconciliation statistics
- ▶ Services
- ▶ Summary of accounts on service
- ▶ Access control information (ACI)
- ▶ Access report
- ▶ Audit events
- ▶ Dormant accounts
- ▶ Entitlements granted to an individual
- ▶ Non-compliant accounts
- ▶ Orphan accounts
- ▶ Policies
- ▶ Policies governing a role
- ▶ Recertification policies report
- ▶ Suspended accounts
- ▶ Separation of duty policy violation report⁵
- ▶ Separation of duty policy definition report⁵
- ▶ User recertification history report⁵
- ▶ User recertification policy definition report⁵

⁵ Tivoli Common Reporting only

One of the available report tabs is shown in Figure 4-38.

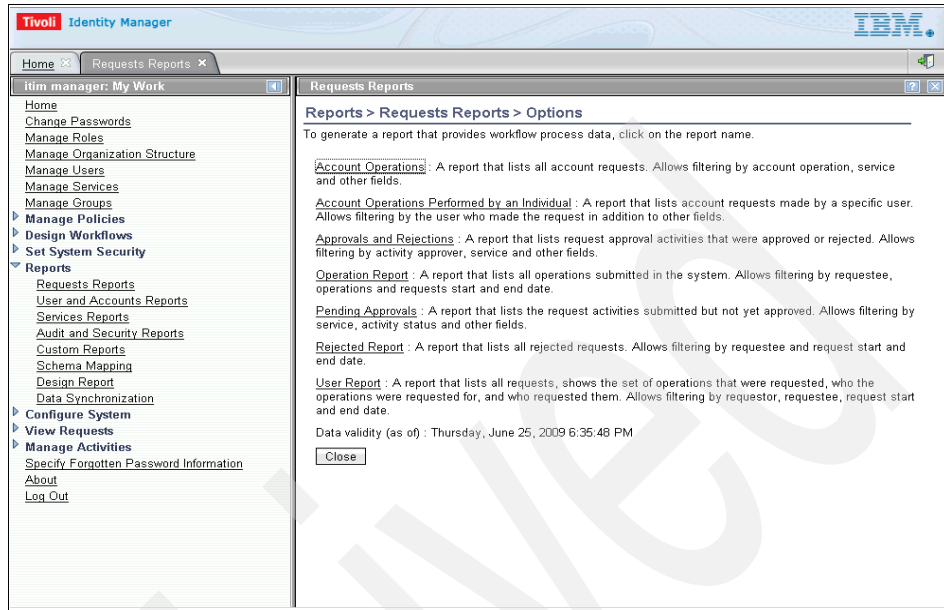


Figure 4-38 Administration console requests reports tab

Figure 4-39 shows a dialog for generating the standard user report.

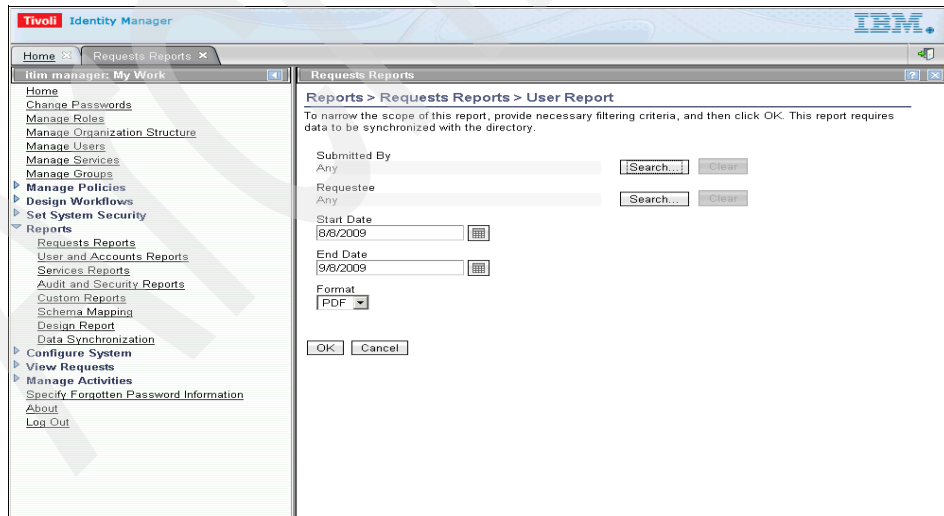


Figure 4-39 Generating a standard user report

Figure 4-40 displays the resulting report.

Tivoli Identity Manager **IBM**

User Report

Report Criteria

Date Printed	17/04/2008
Time Printed	15:12
Time Zone	GMT-05:00
Data validity (as of)	11/04/2008 18:19
Report Generated By	itim manager
Total Entries Processed	89
User Input	Submitted By Like Any
User Input	Requestee Like Any
User Input	Start Date Greater Than 17/03/2008 00:00
User Input	End Date Less Than 17/04/2008 23:59

Request Type	Submitted By	Requested For	Subject	Status	Time Started	Time Completed	Last Access Time
Service Provision Process	System Administrator	Leigh Assauw	lassauw	Aborted	15/04/2008 16:57	15/04/2008 16:58	15/04/2008 16:57
Service Provision Process	System Administrator	Rob Hoffman	rhoffman	Completed	15/04/2008 12:37	15/04/2008 12:37	15/04/2008 12:37
Service Provision Process	System Administrator	Rob Hoffman	rhoffman	Completed	16/04/2008 17:31	16/04/2008 17:31	16/04/2008 17:31
Service Provision Process	System Administrator	Test User0001	0001	Completed	11/04/2008 17:57	11/04/2008 17:57	11/04/2008 17:57
Change Account Process	System Administrator	Test User0001	0001	Completed	14/04/2008 21:26	14/04/2008 21:26	14/04/2008 21:26

© 1999-2007 IBM. All Rights Reserved. Tivoli Identity Manager 5.0 Build 1454 1 of 8

Figure 4-40 User report

Refer to Administering → Report Administration in the IBM Tivoli Identity Manager Information Center Version 5.1 for more details on reporting capabilities, available at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_reports_oview.html

4.3.8 E-mail notification

Tivoli Identity Manager provides external notification of events to users via e-mail. This can range from system events such as account creations to workflow-related events such as an approval. These can be divided into two notification types:

- ▶ System notifications: Do not require user actions. Can be disabled or enabled.
- ▶ Manual activity notifications: Require user actions. Can be completed by performing a manual activity in the recipient's Tivoli Identity Manager account.

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.

Notification templates

Notification templates provide a consistent notification style and content across manual activities and system activities such as adding accounts, changing passwords, approvals, and so on. A standard set of templates is provided with Tivoli Identity Manager, and this can be customized as required. There is also an allowance for text and XHTML to be sent together under different MIME types, and hence be displayed appropriately for the e-mail client used by the recipient. These are all configurable via the Tivoli Identity Manager Web interface and can provide dynamic information via standard documented tags or extended tags defined to a deployment.

The standard system notification templates are:

- ▶ New account template
- ▶ New password template
- ▶ Change account template
- ▶ Restore account template
- ▶ Suspend account template
- ▶ Deprovision account template
- ▶ Activity timeout template
- ▶ Process completion template
- ▶ Process timeout template

The standard manual activity notification templates are:

- ▶ Compliance template
- ▶ Manual activity approval template
- ▶ Manual activity RFI template
- ▶ Manual activity work order template
- ▶ To-do reminder template

The standard recertification notification templates are:

- ▶ Delete account template
- ▶ Mark access template
- ▶ Mark account template
- ▶ Remove access template
- ▶ Suspend account template
- ▶ Access marked template
- ▶ Access removed template
- ▶ Account deleted template
- ▶ Account marked template
- ▶ Account suspended template

- ▶ User recertification pending template

The standard access notification templates:

- ▶ Add access template
- ▶ Remove access template

Refer to Administering → Workflow management → Workflow notification properties in the IBM Tivoli Identity Manager Information Center Version 5.1 for more details on notification templates, at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_wkflo_notify.html

Post office

The post office provides a mechanism for reducing the number of e-mail notifications that 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 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 via the Web interface.

The Group E-mail Topic field in each manual activity definition within a workflow determines similar tasks for the purpose of grouping e-mails, as shown in Figure 4-41. 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. The recipient's preferred locale is honored. This reduces the volume of individual e-mails regarding notifications of the same topic that a user receives.

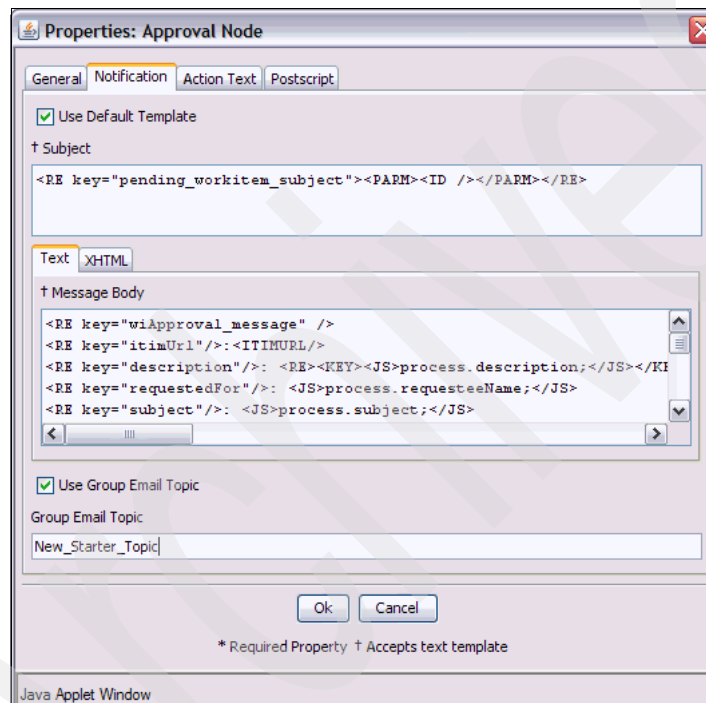
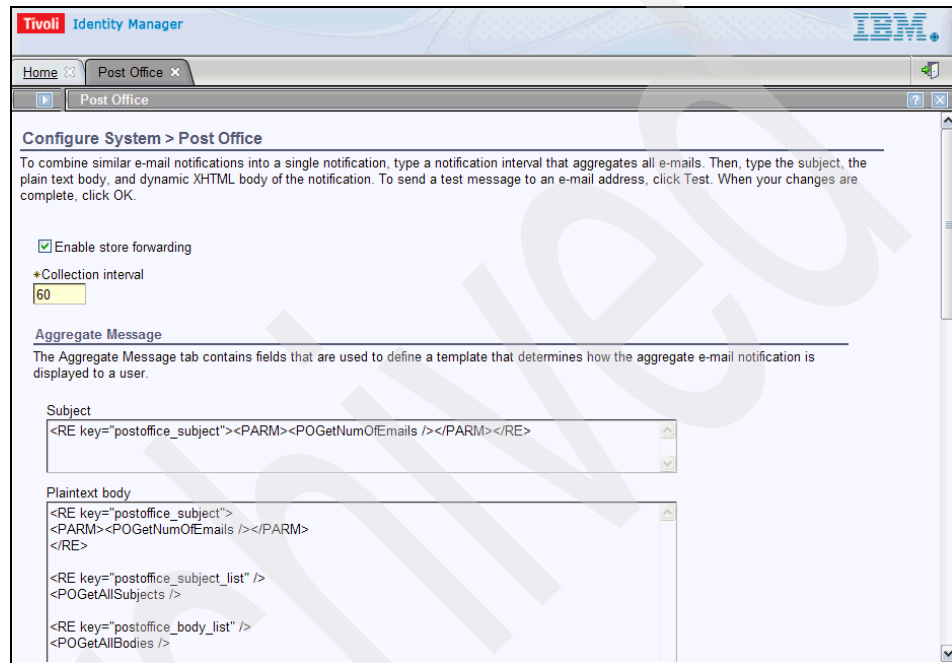


Figure 4-41 Group e-mail topic defined within workflow node

As per the notification template approach specified for the standard notifications, there is an aggregate message tab in the post office configuration section of Tivoli Identity Manager that allows for customization of the template that is used to generate the aggregate message that is sent to the user, as shown in Figure 4-42. As this is a notification template in its own right, it can use dynamic content and can be as simple or as complex as required.



The screenshot shows the Tivoli Identity Manager web interface. The browser tabs include 'Home' and 'Post Office'. The page title is 'Configure System > Post Office'. Below the title, there is a brief instruction: '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.'

There are two main sections:

- Enable store forwarding:** A checkbox that is checked.
- *Collection interval:** A text input field containing the number '60'.

The **Aggregate Message** section contains the following fields:

- Subject:** A text area containing the code: `<RE key="postoffice_subject"><PARM><POGetNumOfEmails /></PARM></RE>`
- Plaintext body:** A text area containing the code: `<RE key="postoffice_subject"><PARM><POGetNumOfEmails /></PARM></RE>`
`<RE key="postoffice_subject_list" /><POGetAllSubjects />`
`<RE key="postoffice_body_list" /><POGetAllBodies />`

Figure 4-42 Post office aggregation notification template

4.3.9 Manage activities

The activities management sections are where an Tivoli Identity Manager user views and completes *action items* that have been assigned to him. Action items listed in a user's activities list are part of workflow processes that require the specified user's participation before they can complete. These action items can be individual or grouped approvals, requests for information, work orders, or compliance alerts. Approvals, work orders, requests for information, or compliance alerts that are grouped are approved, rejected, or submitted as one single unit.

Figure 4-43 shows approver Dan Meyers' activities list, which includes a separation of duty rule violation approval and a reconciliation approval in the administrative console.

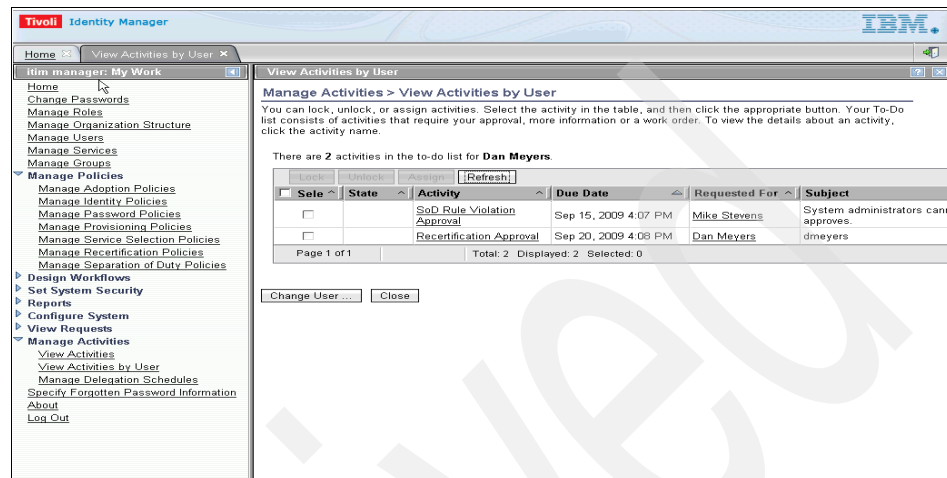


Figure 4-43 Activities list for Dan Meyers in the administrative console

Figure 4-44 shows the same list from the self-care interface.

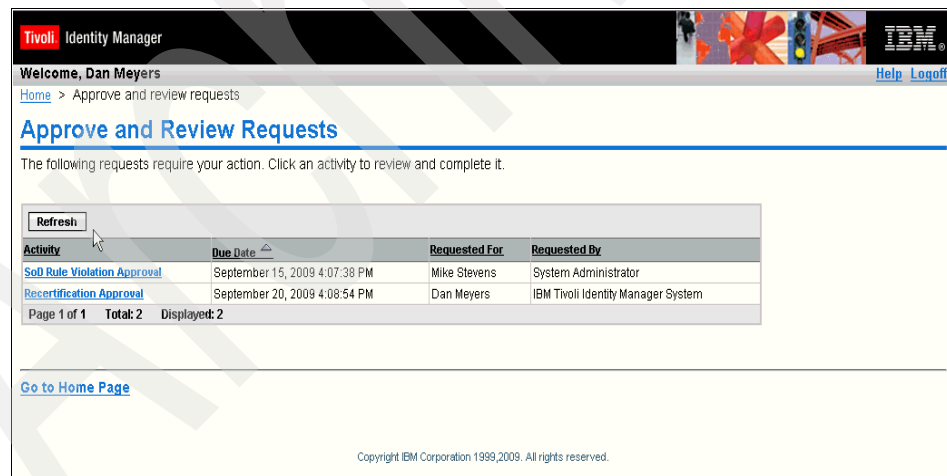


Figure 4-44 Activities list for Dan Meyers in the self-care interface

The to-do list page can contain the following item types:

- ▶ Approval requests, including recertification approvals
- ▶ Work order requests
- ▶ Requests for information
- ▶ Policy compliance alerts

Locking, unlocking, and forwarding to-do list items

The Tivoli Identity Manager administrative console's activities list page allows a user to lock, unlock, assign, and respond to activities. Locking an activity disables the ability for other authorized users of the activity to action the request. This removes the potential for conflicts to arise due to multiple users working on the same activity. Unlocking an activity returns it to a state that allows other authorized users to action the request. Assigning an activity can be thought of as locking the activity for another user. For example, if user A forwards the activity to user B, it is locked for user B to act upon. A different icon is displayed to visually differentiate a forwarded item from a locked item that has been locked by a user for themselves, as shown in Figure 4-45 and Figure 4-46 on page 180.

Tivoli Identity Manager

View Activities x

View Activities

Manage Activities > View Activities > Grouped Approval

Approve/Reject the Request

<input type="checkbox"/>	Se...	State	Due date	Requeste...	Subject	Request t...	Service N...
<input type="checkbox"/>			01-May-2008 22:24	Leigh Assauw	lassauw	Account Add	TAAir Active Directory service
<input type="checkbox"/>			01-May-2008 22:26	Charles Doddy	cdoddy	Account Add	TAAir Active Directory service

Page 1 of 1 Total: 2 Displayed: 2 Selected: 0

Comments

Close

Figure 4-45 Locked activity

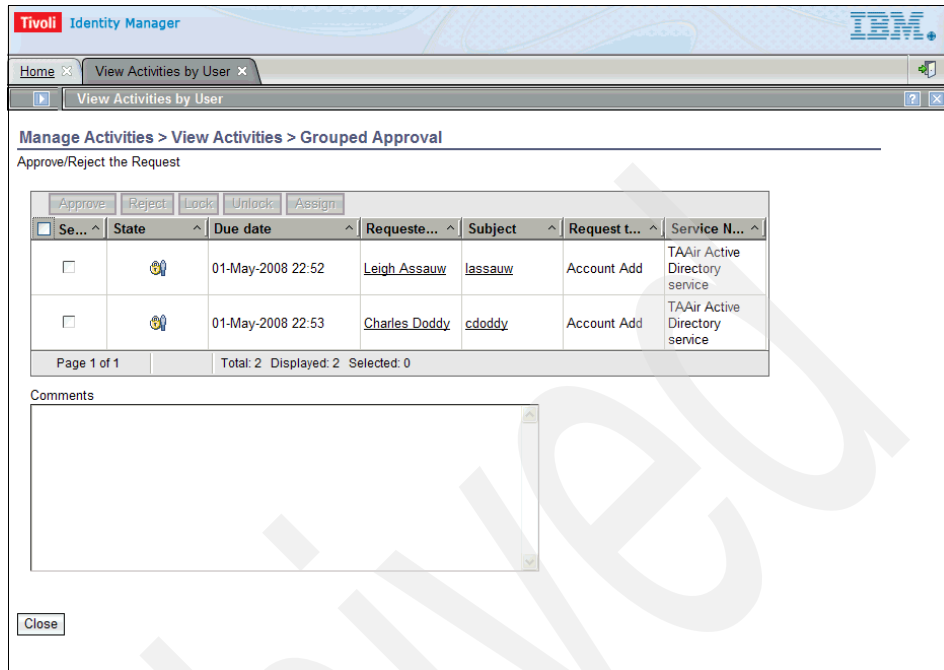


Figure 4-46 Assigned activities

System administrators can view all users' activities and respond to any action item in any user's activity list. However, system administrators cannot lock work items unless they meet the criteria of being a potential owner of the item.

Delegate function

Users can select a delegate to action their activities, in particular, the approval requests and the request for information to-do items, as shown in Figure 4-47 on page 181. This is useful in the event that a user knows she will be away on leave or for various other reasons that prevent her from having access to Tivoli Identity Manager. More than one delegate can be selected for a person, but more than one delegate cannot be selected for the same time period. To change the delegate for a specific time period, the originally selected delegate must be deleted and a new delegate must be added for that time period. In our example below user Rob Hoffman (rhoffman) is delegating workflow-related actions to Leigh Assauw (lassauw) between the dates shown.

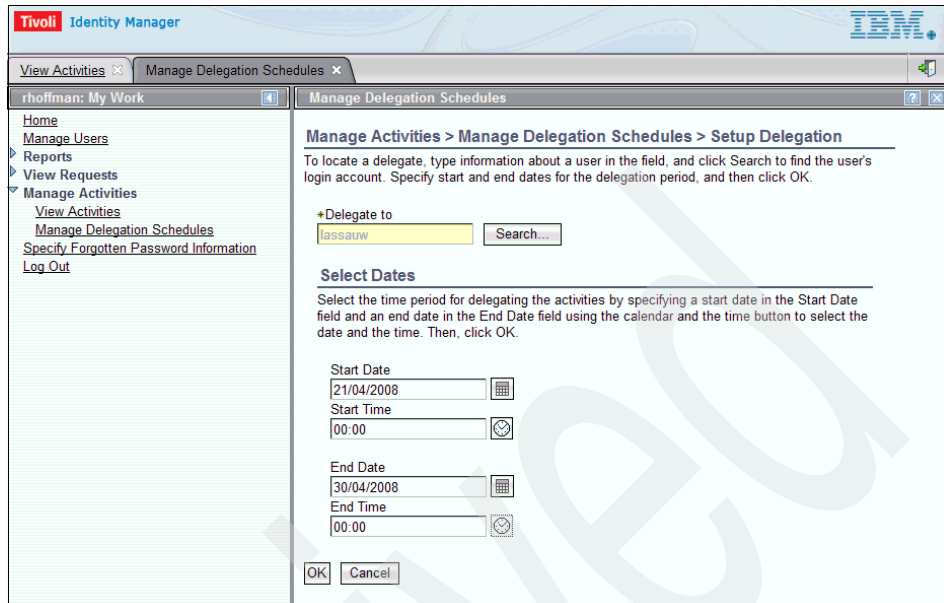


Figure 4-47 Delegating workflow-related actions

4.3.10 Import/export

Many enterprise applications, including Tivoli Identity Manager, are often deployed in stages. New policies and business logic can be developed and tested in a particular environment and then migrated to another similar environment. 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 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 following types of data objects can be exported from Tivoli Identity Manager using these features:

- ▶ Adoption policies
- ▶ Group
- ▶ Identity policies
- ▶ Tivoli Identity Manager groups
- ▶ Life cycle operations
- ▶ Life cycle rules
- ▶ Child role
- ▶ Password policies
- ▶ Provisioning policies
- ▶ Services
- ▶ Service selection policies
- ▶ Workflows

Export

The data objects can be selectively exported or an administrator can specify that a full export is to be performed. That is, an administrator can perform a partial export or an entire export. The exported data is archived into a Java Archive (JAR) file, which is stored within the Tivoli Identity Manager relational database. The administrator can download this JAR file to their local system for import into another similar Tivoli Identity Manager environment. This functionality can also be used as a way of keeping system configuration levels to roll back to in the case of a configuration error. In other words, it can assist with Tivoli Identity Manager configuration management and version control in addition to serving its primary purpose of allowing for migration of data between environments.

In order to guarantee the integrity of the data throughout the migration process, Tivoli Identity Manager automatically detects and includes any dependencies that an exported object may 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.

Exporting everything through the use of the full export saves all of the data supported by the export in the system. Performing a partial export of individual items might not 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 instance, when exporting a provisioning policy that includes an automatic account creation function, the identity policy needed to create the user ID will not be exported as a dependency of the provisioning policy, because it is not required for the creation of the provisioning policy. However, it could be required for the execution intended for the provisioning policy and, if so, it should be exported and imported as a separate object.

Figure 4-48 illustrates an example provisioning policy being exported.

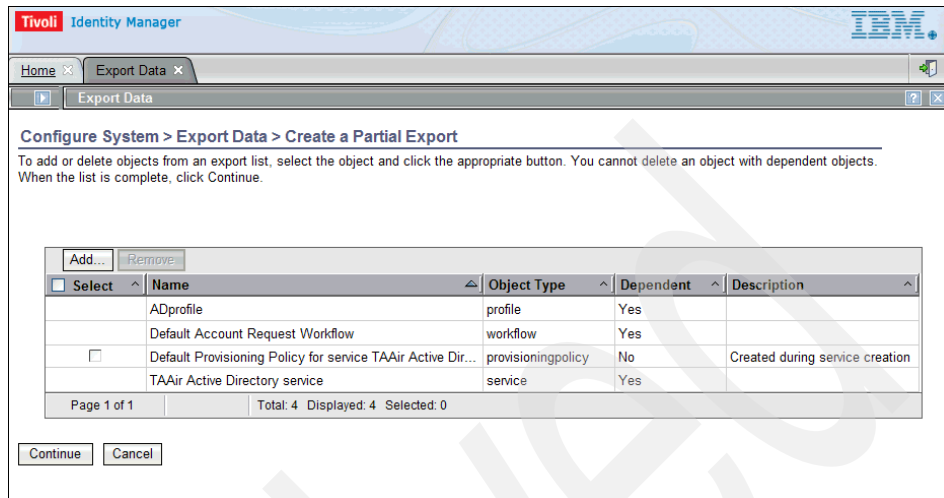


Figure 4-48 Export of a provisioning policy

Import

The import process is initialized by an administrator on a target server after extracting objects (generating an export JAR) from a source server. The import process consists of several individual stages that allow the administrator to interact with and configure the process:

- ▶ Jar file upload
- ▶ Difference evaluation
- ▶ Conflict resolution
- ▶ Data commit to the system

Once 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 once the conflicts are resolved by the administrator. Note that importing provisioning policies and dynamic organizational roles might trigger a sub-stage in the import process that involves associating different people with new roles and triggering policy enforcement if imported policies have changes that require re-evaluation.

Figure 4-49 shows a previously exported JAR file being imported into a different Tivoli Identity Manager environment.

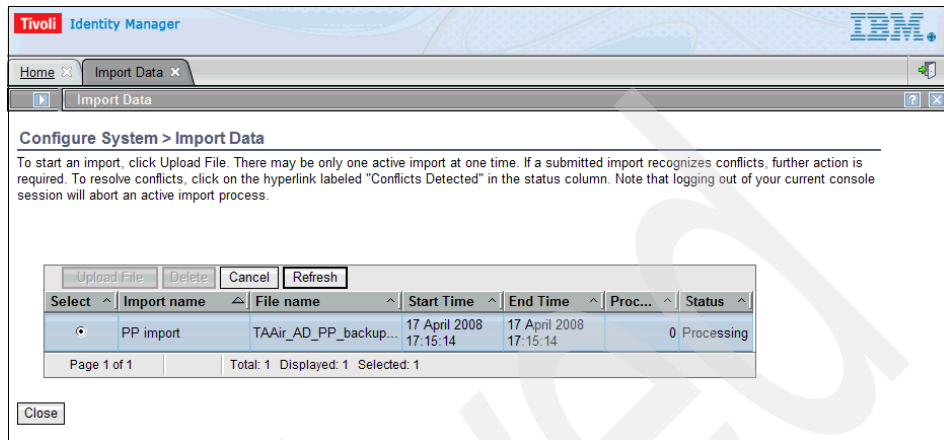


Figure 4-49 Import of previously exported file

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.

After ensuring that 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 organization chart 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 re-establish 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 must 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 re-establish the link for those objects. The target server's trace.log can be examined for helpful troubleshooting messages in cases where there are differences in the organizational structure of a source and a target server or if dependencies are not found on the target. Organizational structure mismatch can cause the import to fail and might require the import to be re-run.

Service profile installation

Tivoli Identity Manager provides administrators with a Web-based interface via the import/export function to import a service profile using a JAR file. This loads all the requisite information into Tivoli Identity Manager required to define a service instance for the profile in question. For example, loading a Windows Active Directory service profile allows for the definition of Windows Active Directory services whose accounts can then be managed by Tivoli Identity Manager. Note that it is possible to package more than one service profile into a single JAR file.

4.3.11 Managing groups

Tivoli Identity Manager has centralized the management of groups into a dedicated administrative interface. This brings together the management of Tivoli Identity Manager groups and groups located on managed resources, such as *Active Directory* and *UNIX* groups, in one logical location.

Where group management is supported by the adapter (see the Adapter Configuration sections in the IBM Tivoli Identity Manager Information Center Version 5.1 for more details), you may create, delete, and modify groups.

This feature allows administrators to manage groups byway of Identity Manager and benefit from the ease of use of the GUI and Identity Manager auditing and reporting facilities. This can enable tasks previously executed by various system administrators specialized in the specific operating system or applications to be carried out by centralized administration staff or automated provisioning policies.

Before you can manage groups, the service must exist within Tivoli Identity Manager, and you must have carried out at least one reconciliation to discover any existing groups. Figure 4-50 shows the **Manage Groups** → **Select a Service** window.

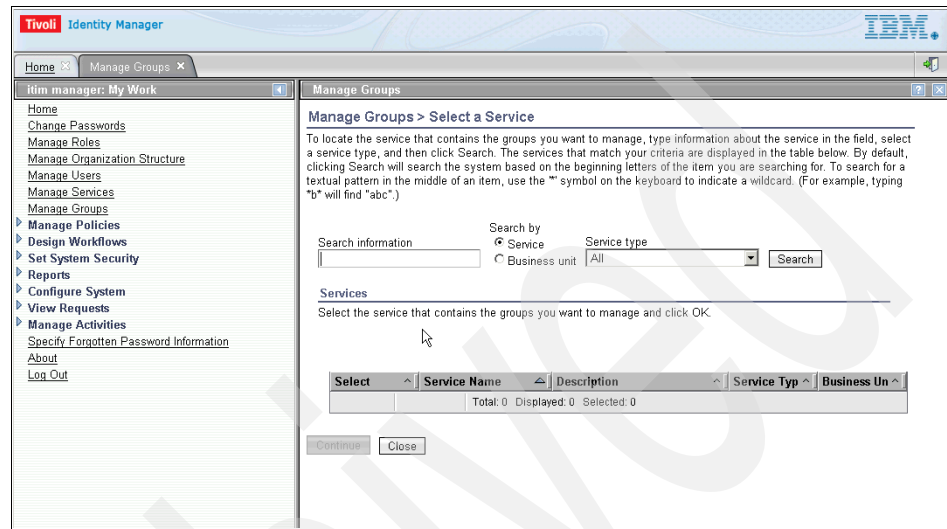


Figure 4-50 Manage groups: Select a Service

Once a service has been selected, either by searching for it by name or clicking the radio button next to the service name after a wildcard search, click **Continue**. This opens the **Manage Groups** → **Select Group** options, as shown in Figure 4-51 on page 187. From here you may search for existing groups and modify or delete them. You may also create new groups.

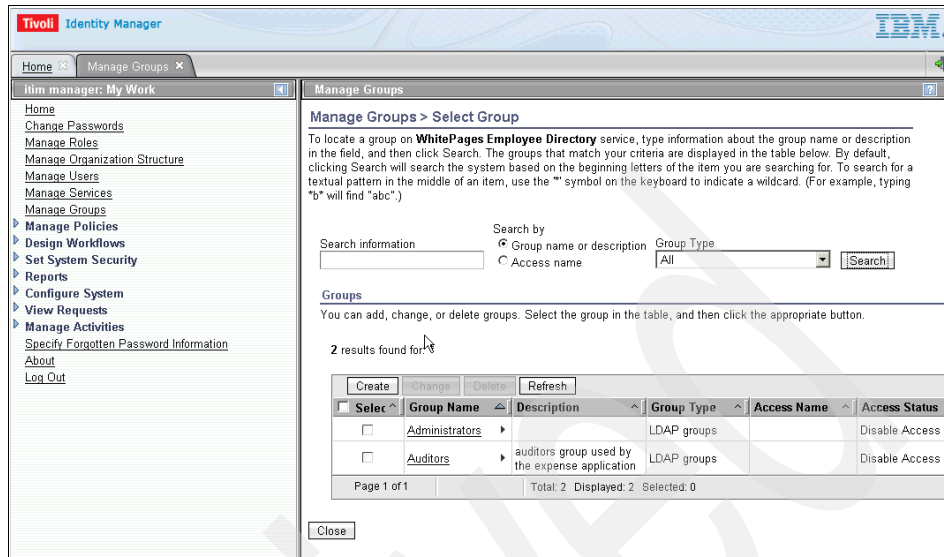


Figure 4-51 Manage groups

A major benefit over previous versions of Tivoli Identity Manager is that once the group has been created, it is immediately available for use within Identity Manager. Previously the group had to be created natively on the targeted managed resource by a local system administrator for that resource. Then the Identity Manager administrator was required to run a reconciliation before the group was available for use.

This group management capability in Tivoli Identity Manager can reduce the local knowledge needed to create and maintain groups and it can reduce the overall time to productivity in group management tasks.

4.4 User interface and access control

In this section we focus on levels of administration, levels of self-care, controlling the scope of administration, and managing access to Tivoli Identity Manager's interfaces and functionality.

Tivoli Identity Manager provides two interfaces, one for administration and one for self-care, both accessible via a Web browser. We describe how to manage permissions and access to views to suit the needs of enterprise user groups.

Administrator roles, Tivoli Identity Manager groups, views, and ACIs

Tivoli Identity Manager permissions are attached to Tivoli Identity Manager users through Tivoli Identity Manager groups. These groups are assigned to Tivoli Identity Manager accounts, which are used to log on to the application and perform management tasks. Tivoli Identity Manager groups grant access to resources, including person records, accounts, configuration items (such as services and workflows), and reports.

More than one Tivoli Identity Manager user can be assigned to an Tivoli Identity Manager group. Permissions are cumulative, so that the total permissions assigned to an Tivoli Identity Manager user correspond to the sum of the permissions granted to that user by each Tivoli Identity Manager group of which he is a member.

In the following example we show how permissions can be assigned in Tivoli Identity Manager. An organization has a central head office within a central IT support group and help desk staff taking calls from users. From an administration perspective, the following business roles have been identified:

- ▶ Users are only allowed to view their own information, request accesses that have been defined for them, and reset their own password.
- ▶ Help desk operators can view other people's information, reset passwords, and view and modify user information.
- ▶ The IT support group has two identified levels of administrators within the organization, which are:
 - Junior administrators, who have the same functionality as the help desk operators, with the ability to create users
 - Senior administrators, who have full control and are the approvers for all workflow processes

From these requirements, it can be inferred that users only must perform basic self-care operation in Tivoli Identity Manager. Help desk operators need access to some administrative capabilities, matching their needs. The same goes for junior administrators. Senior administrators have access to most, if not all, administrative capabilities of Tivoli Identity Manager. Table 4-1 on page 189 provides a summary of these accesses.

Table 4-1 An example of users mapped to roles and permissions

Job role	Department	Access rights	Views
Senior administrator	IT support	<ul style="list-style-type: none"> ▶ Full control ▶ Workflow approvers 	<ul style="list-style-type: none"> ▶ All administration console views ▶ All self-care views
Junior administrator	IT support	<ul style="list-style-type: none"> ▶ View all person information ▶ Create person ▶ Manage person accounts ▶ Manage person accesses 	<ul style="list-style-type: none"> ▶ Limited administrative console access ▶ Full self-care interface access
Help desk operations	Help desk	<ul style="list-style-type: none"> ▶ View all person information ▶ Manage person accounts ▶ Manage person accesses 	<ul style="list-style-type: none"> ▶ Limited administrative console access ▶ Limited self-care interface access
Users	Various	<ul style="list-style-type: none"> ▶ View self information ▶ Request accesses for self 	Limited self-care interface access

Business roles and corresponding permissions can now be implemented.

Access control items (ACIs)

An access control item is a configuration item that defines a set of permissions for a type of information managed by Tivoli Identity Manager. This type of information can be a service, a type of person record, or a specific report. Aside from Tivoli Identity Manager system administrators, users have no access to Tivoli Identity Manager entities unless an ACI grants permission.

Permissions are set on an object at various levels:

- ▶ Where on the organizational tree the permission is set and with which scope
- ACIs are configured in organizational tree containers. The permissions defined in the ACI can be set to apply to view the container in which it was defined or to the container and all containers below it.

- ▶ To what the permissions apply

ACI permissions are applied to Tivoli Identity Manager entity types, such as all service types, all person types, and all organizational roles. Where applicable, these can be further refined to specific entities, like a specific person entity or a specific service. Additionally, a filter can be defined to apply permissions to a subset of the entity types and entities specified.
- ▶ Which global operations can be performed on the object as a whole

A number of operations can be performed on all Tivoli Identity Manager entities, although these operations vary from entity type to entity type. ACIs can grant permission to individual operations like add, modify and delete.

Available permissions are none, grant, and deny. None does not grant or remove any permissions. Grant grants permission to the user to access the defined criterion, and overrides none when cumulating permissions. Deny removes permission to access the defined criterion and overrides both none and grant when cumulating permissions.
- ▶ Which object attributes can be read or updated

All Tivoli Identity Manager entities contain attributes. ACIs define read and write permissions for all attributes belonging to an entity or all attributes common to all entities of a defined entity type.

Each attribute is granted both a read and a write permission, each of which can be configured with a none, grant, or deny value. None does not grant or remove any permissions. Grant grants permission to the user to access the defined criterion and overrides none when cumulating permissions. Deny removes permission to access the defined criterion and overrides both none and grant when cumulating permissions.
- ▶ The group of users or principal to which the permissions apply

Permissions are applied to Tivoli Identity Manager account users. ACI members can be defined in two ways:

 - By choosing a number of principals from the available list defined in the ACI, such as *all account holders*, *the manager of the account owner*, or *the owner of the organizational role*. Principals vary depending on the entity type on which the ACI grants permission.
 - By adding Tivoli Identity Manager groups as members of the ACI. The ACI then applies to those group members.

The following figures illustrate how ACIs are defined. They represent how, according to our previous example, an ACI can be customized for help desk operators to modify and search for person records in Tivoli Identity Manager. Figure 4-52 illustrates how general ACI configuration options are set. The content of the following tabs depend on these selections.

The screenshot shows the 'Set System Security > Create Access Control Item > General' configuration page in Tivoli Identity Manager. The page has a left-hand navigation pane with tabs for 'General', 'Operations', 'Permissions', and 'Membership'. The 'General' tab is selected. The main content area contains the following fields and options:

- Name:** A text field containing 'Help_Desk_User_ACI'.
- Protection Category:** A dropdown menu set to 'Person'.
- Type:** A dropdown menu set to 'inetOrgPerson'.
- Apply object protection on this business unit:** A section with a search box containing 'Tivoli Austin Airlines' and a 'Search...' button. Below it, the checkbox 'all of its sub units' is checked.
- Apply protection to...:** Two radio button options: 'All objects in the selected category or class' (which is selected) and 'A subset of objects that satisfy the filter criteria'. Below these options is a large, empty text area for filter criteria.

At the bottom of the page, there are four buttons: 'Back', 'Next >', 'Finish', and 'Cancel'.

Figure 4-52 ACI general configuration

Figure 4-53 shows how global operations for the selected entity or entity type are set. In this case only the modify and search operations are granted.

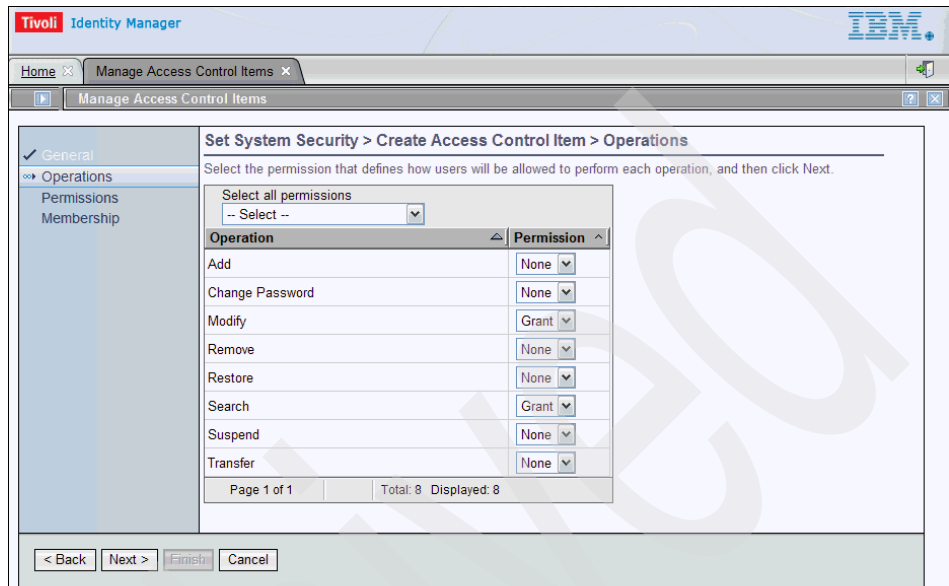


Figure 4-53 ACI operations configuration

Figure 4-54 shows how permissions are set for individual entity or entity type attributes. In this case read permission is granted for all attributes aside from the driver license attribute. Write permission has not yet been granted to attributes.

Set System Security > Create Access Control Item > Permissions

Select the permission that defines how users will be allowed to perform each attribute, and then click Next.

Select all read: -- Select -- Select all write: -- Select --

Attribute	Read	Write
Administrative assistant	Grant	None
Aliases	Grant	None
Audio	Grant	None
Business category	Grant	None
Department number	Grant	None
Description	Grant	None
Destination indicator	Grant	None
Display name	Grant	None
Driver license	None	None
E-mail address	Grant	None
Employee number	Grant	None
Employee type	Grant	None
Fax number	Grant	None
First name	Grant	None
Full name	Grant	None
Home address	Grant	None
Home telephone number	Grant	None
Initials	Grant	None
International ISDN code	Grant	None
Jpeg Photo	Grant	None
Last name	Grant	None
Last Operation	Grant	None

Figure 4-54 ACI permissions configuration

Finally, Figure 4-55 shows how user principals and Tivoli Identity Manager groups can be defined as ACI members. here only the HelpDesk Operators group has been granted membership of to ACI.

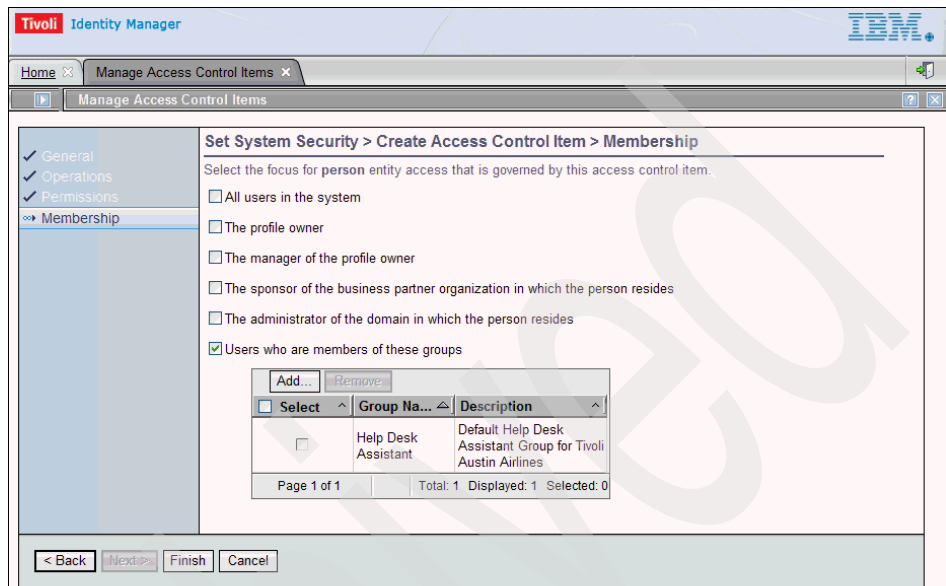


Figure 4-55 ACI membership configuration

Tivoli Identity Manager views

Tivoli Identity Manager *views* define access to most Tivoli Identity Manager interface tasks in both the administration console and the self-care interface. They are managed from the administrative console. Individual views are managed by selecting interface tasks in the menu provided, as shown in Figure 4-56, which represents the HelpDesk view. Notice how only relevant tasks selected relevant to searching for and modifying users, as well as self-care, are selected.

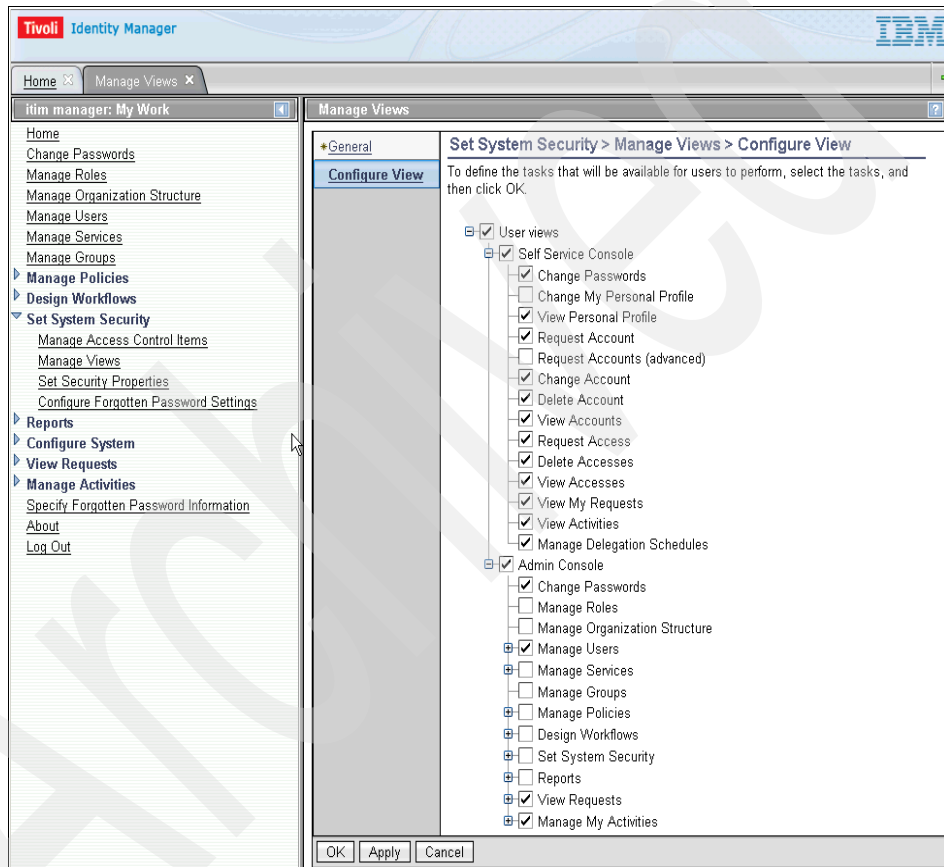


Figure 4-56 View configuration

Views are assigned to Tivoli Identity Manager groups in the group option window, which means that views must be created prior to being assigned to roles. Only one view can be defined per group, as shown in Figure 4-57, although individual views can be assigned to several groups. Groups require a view to be assigned to them.

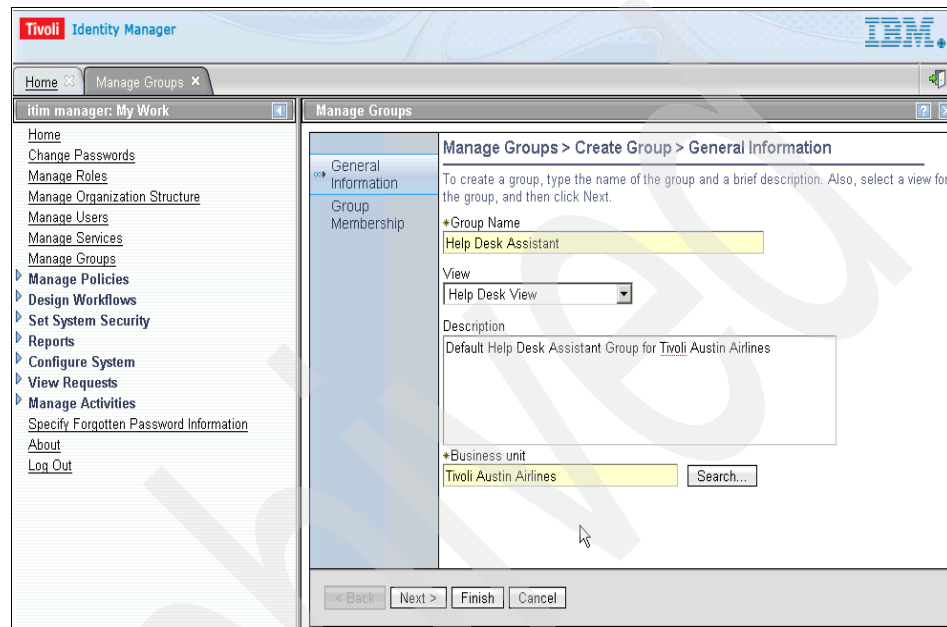


Figure 4-57 Help desk group with view defined

4.5 Tivoli Identity Manager schedules

Tivoli Identity Manager is normally used in real time or near-real time. However, there are some time-based aspects to Tivoli Identity Manager:

- ▶ Scheduling of changes: Many changes made through the administrative console can be scheduled to occur at a specific date and time.
- ▶ Scheduling of reconciliations: The reconciliation process can be scheduled to occur periodically.
- ▶ Time limits on workflow: Workflows are by nature asynchronous. The processes can be configured to time out and escalate to ensure that changes do not hang. Also, a to-do list reminder could be sent by e-mail after a configured amount of time.
- ▶ Recertification policies: Recertification is an event occurring either at specific calendar intervals or on a rolling calendar basis.

- ▶ Life cycle rules: Define events that are triggered based on a time interval or on time and matching criteria.
- ▶ Post office: The post office will send e-mail notifications based on message (collect similar notifications), user, and a period of time.
- ▶ Historical reporting: Most reporting is historical.

4.5.1 Scheduling of changes

Most administrative changes in the Tivoli Identity Manager administrative console can be scheduled. Actions submitted in the self-care console are performed immediately.

Administrative console users with appropriate access can schedule a password change. Figure 4-58 shows a password change scheduled to occur at 11.17 hours on 8 Sep 2009.

The screenshot shows the Tivoli Identity Manager administrative console. The main content area is titled "Change Passwords" and contains the following elements:

- Instructions:** "To change the password for **Mark Foster**, select whether you want to have the system generate the password or whether you want to specify the password now. If you specify a password, it must conform to the rules for the password for this account. To view these rules, click [View password strength rules](#)."
 - Generate a password for me
 - Allow me to type a password
- Form fields:** "Password" and "Confirm Password" fields, both containing masked text (*****).
- Links:** [View password strength rules](#)
- Accounts:** A section titled "Accounts" with the text "Your password will be changed for the accounts listed in the table below."

Service Name	User ID
ITIM Service	mfoster
WhitePages Employee Directory	mfoster

Page 1 of 1 Total: 2 Displayed: 2
- Schedule:**
 - Immediate
 - Effective date
 - Date: 9/8/2009
 - Time: 11:17 AM
- Buttons:** "Submit" and "Cancel"

Figure 4-58 Scheduling a password change

Most of the administrative functions can be scheduled in the administrative console, including:

- ▶ User (person): Add, modify, suspend, resume, transfer, password reset, and delete.
- ▶ Account: Add, modify, suspend, restore, password reset, and de-provision.

Figure 4-59 shows a suspend action against a person.

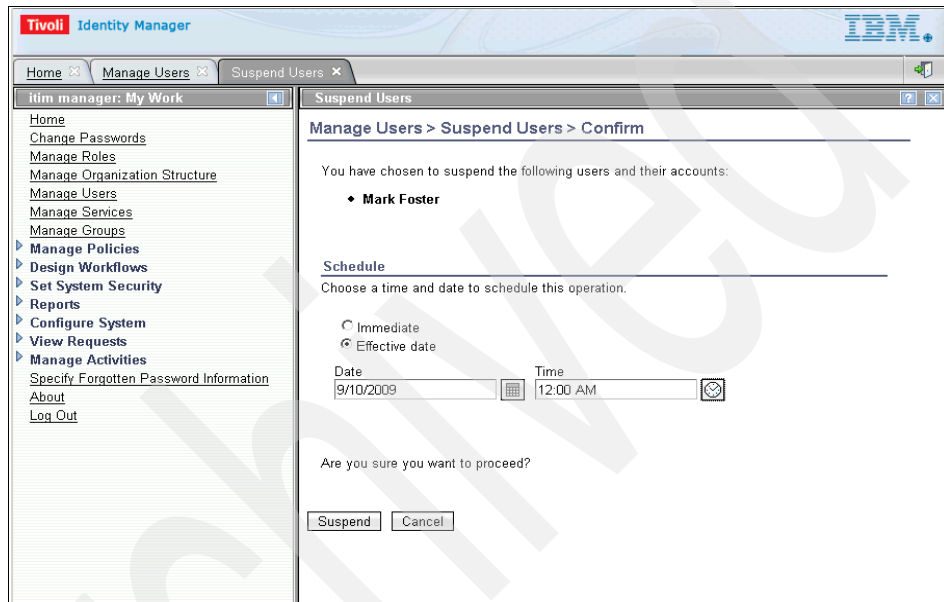


Figure 4-59 Scheduling a user suspension in the administrative console

All scheduled actions show up in the requestor's pending requests list. Figure 4-60 shows a pending suspension request.

The screenshot displays the Tivoli Identity Manager administrative console. The page title is "View Requests > View All Requests". Below the title, there are search filters for "Request type" (set to "All") and "Time Interval" (set to "Today"). There are "Search Requests" and "Reset" buttons. A link for "More Search Criteria" is also present. The main content area shows "1 requests were submitted between 18 April 2008 and 18 April 2008." Below this, there is a table with columns: "Sel...", "Status", "Request type", "Date submitted", "Requestor", "Requested for", and "Service Name". The table contains one row with a checkbox, a status of "Pending", a request type of "Suspend User", a date of "18 April 2008 13:18:46", a requestor of "System Administrator", and a requested for user of "Charles Doddy". There are also "Cancel request" and "Refresh" buttons above the table, and a "Close" button at the bottom.

Sel...	Status	Request type	Date submitted	Requestor	Requested for	Service Name
<input type="checkbox"/>	Pending	Suspend User	18 April 2008 13:18:46	System Administrator	Charles Doddy	

Figure 4-60 Pending changes viewed from the administrative console

As shown in Figure 4-60, a request that has been scheduled can be aborted, but the scheduled date and time cannot be changed.

4.5.2 Scheduling of reconciliations

Reconciliations should be scheduled to run periodically. The aim of the reconciliation is to determine any differences between the account information in Tivoli Identity Manager and the account information about a managed system or application (service).

Multiple reconciliations of individual services can be scheduled. You can schedule reconciliations to run on an hourly, daily, weekly, or monthly basis. Figure 4-61 shows how to modify a reconciliation schedule.

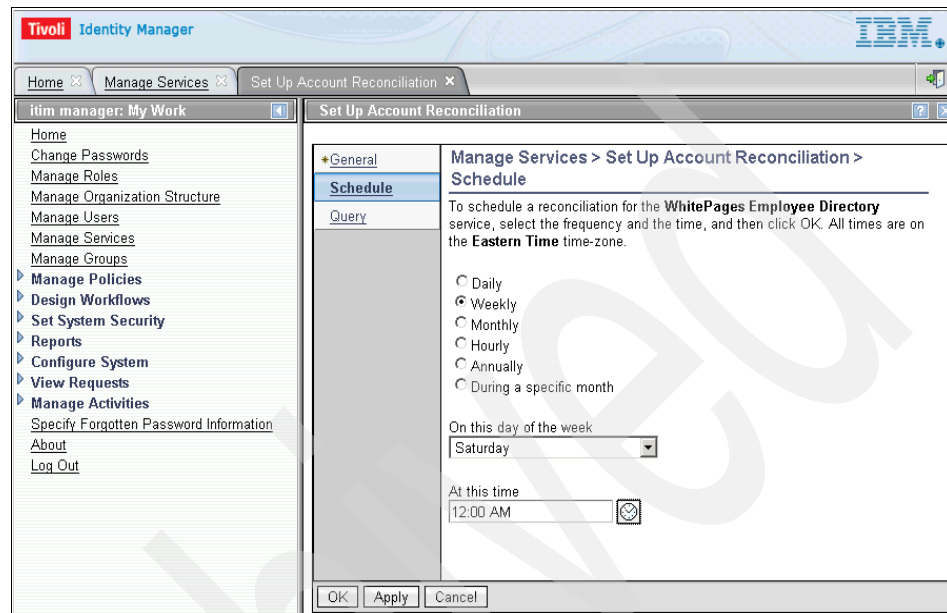


Figure 4-61 Scheduling reconciliations

Administrators should also periodically check the orphaned account list for the service and process the orphaned accounts.

4.5.3 Time limits on workflow

Workflow is an asynchronous mechanism. For example, when a request is raised that triggers workflow, e-mails are sent to approvers who log into Tivoli Identity Manager and approve or reject the request. To ensure that a request does not languish in a workflow step waiting for an approver who is on leave, workflow has two mechanisms for timeouts:

- ▶ Escalation limits

Where an escalation approver can be specified and, if the first approver does not act on the request, the request is bumped up to the escalation participant.

- ▶ Loops

A step or set of steps in a workflow can repeatedly loop a set number of times (with the escalation limits applying to the steps) and then expire.

Figure 4-62 shows the approval node dialog with escalation limits.

The screenshot shows a dialog box with four tabs: General, Notification, Action Text, and Postscript. The General tab is active. The fields are as follows:

- * Activity ID: Manager_Approval
- Activity Name: Manager Approval
- * Participant: Manager (dropdown)
- Escalation Participant: System Administrator (dropdown)
- Escalation Limit: 14 Days, 0 Hours, 0 Minutes, 0 Seconds
- Join Type: AND OR
- Split Type: AND OR
- Entity Type: Account (dropdown)
- Input Parameters: Search Relevant Data (button)

ID	Type	Relevant Data ID
entity	Account	entity
service	Service	service
owner	Person	owner

Buttons: Ok, Cancel

* Required Property † Accepts text template

Figure 4-62 Setting escalation limits in workflow

Note that it is possible to define time limits for both simple and advanced workflow activities.

Activities reminder

Tivoli Identity Manager can also be configured to send activity notifications and 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.

Using a defined reminder interval, a reminder e-mail will be sent to participants in workflows for activities waiting for their action.

Figure 4-63 shows a time period configuration for activities.

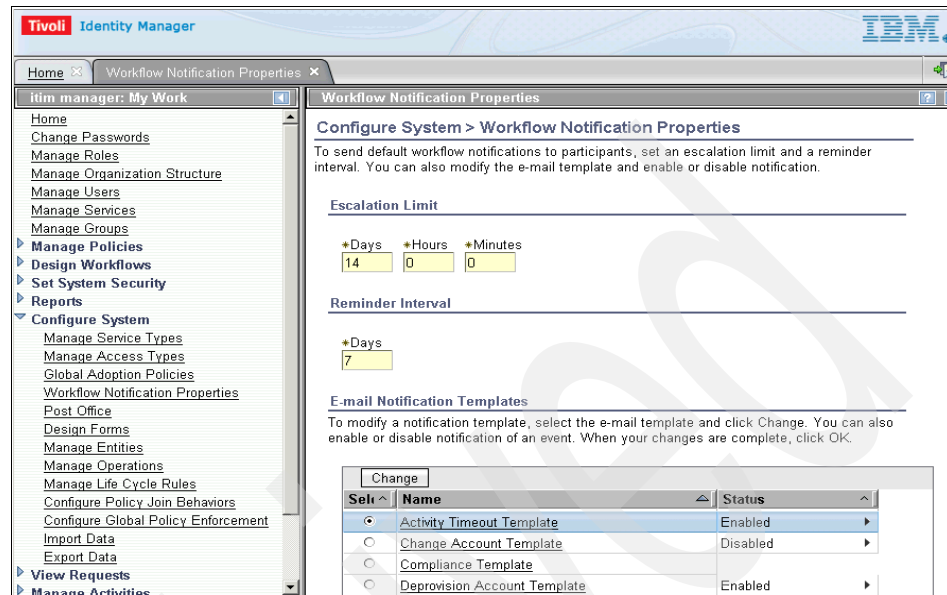


Figure 4-63 Reminder interval

4.5.4 Recertification policies

Recertification policies allow users, accounts, and accesses to be recertified periodically or on a rolling calendar basis. Whenever an users, account or access is due for recertification, a workflow defining the recertification process is executed.

Figure 4-64 shows how recertification scheduling is configured.

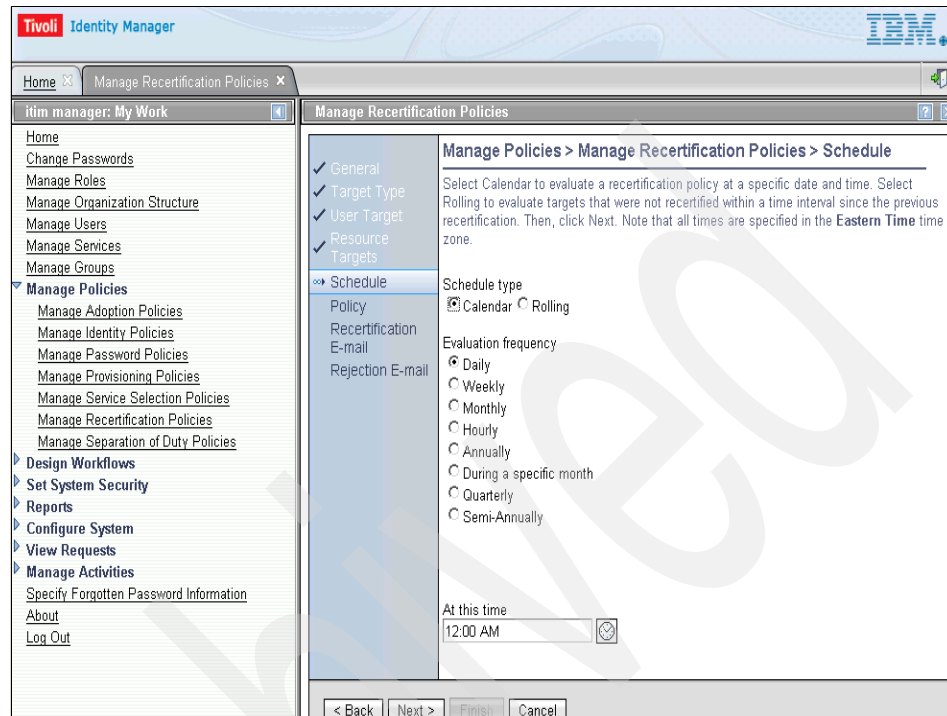


Figure 4-64 Recertification scheduling

4.5.5 Life cycle rules

Life cycle rules allows administrators to define events that can be triggered based on time or based on time and matching criteria evaluated against an entity. Those can be defined as global rules (based on time) or as entity or entity type rules (based on time and matching criteria).

For global rules, an event is defined by a time interval, for example, once a month, or every Monday at 8:00 a.m. Global life cycle rules are independent of any particular system entity. The life cycle operations that can be invoked by a global rule must also be global in nature, as there will be no context available to call an entity or entity-type-based operation.

Entity and entity-type rules also use events based on a time interval. However, the goal of these rules is to affect multiple life cycle objects at one time. Therefore, a separate event is triggered for each life cycle object. To keep events from triggering possibly thousands of objects that might not be related to the rule, a matching criteria is available for these rules. Without the matching criteria, every object of the given entity or entity type will have the associated life cycle operation performed on it.

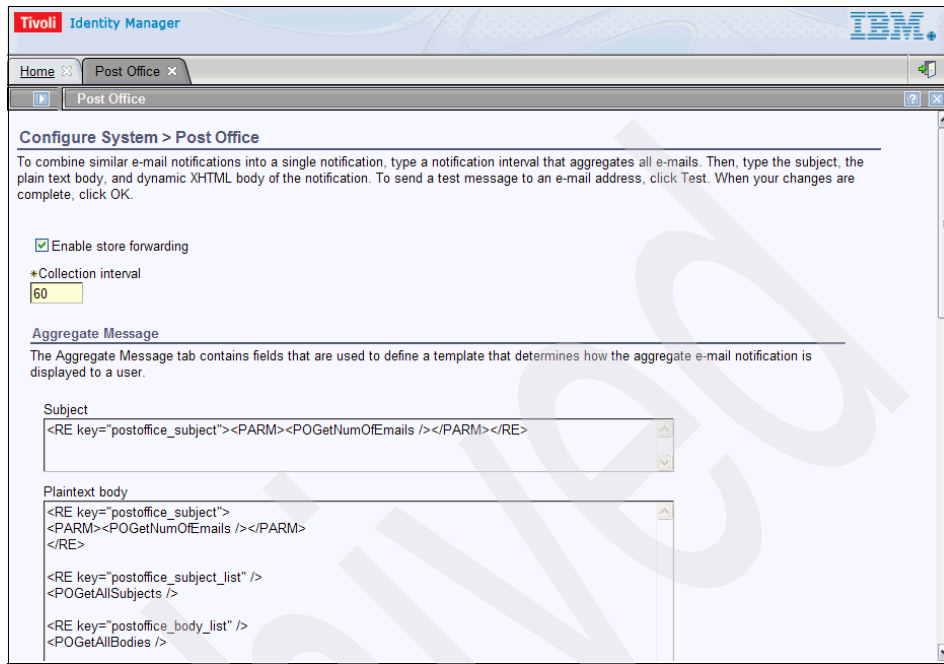
You can assign a time period for a life cycle. Possible values are hourly, daily, weekly, monthly, and an arbitrary list of dates that recur on an annual basis. The field following the time period changes to an appropriate user interface element when the time period select widget is changed:

- Once** The date and time to trigger the event and a check box for immediately.
- Hourly** The minutes past the hour.
- Daily** The 24-hour time period to start at in hours and minutes.
- Weekly** The day of the week and time of the day.
- Monthly** The day of the month and time of the day.
- Annually** The month, the day, and the time of day. Multiple dates can be specified in the rule to support bi-annual and quarterly dates.

4.5.6 Post office

The post office is used to reduce the number of e-mail notifications that a user could receive. It collects similar notifications for a period of time and once the interval expires e-mail notifications are combined into a single e-mail and forwarded to activity participants.

The period of time is configured in minutes, as shown in Figure 4-65.



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 />

Figure 4-65 Post office configuration

4.5.7 Historical reporting

Most reports available through the Report menu item can be configured to limit the time frame for a report. For example, Figure 4-66 shows the parameters for running an operation report for the system administrator creating new users over a 24-hour period.

The screenshot shows the Tivoli Identity Manager web interface. The left sidebar contains a navigation menu with categories like 'Home', 'Manage Roles', 'Manage Users', 'Manage Policies', 'Design Workflows', 'Set System Security', and 'Reports'. The 'Reports' category is expanded, showing sub-items like 'Requests Reports', 'User and Accounts Reports', 'Services Reports', 'Audit and Security Reports', 'Custom Reports', 'Schema Mapping', and 'Design Report'. The main content area is titled 'Requests Reports' and 'Reports > Requests Reports > User Report'. It includes a sub-header: 'To narrow the scope of this report, provide necessary filtering criteria, and then click OK. This report requires data to be synchronized with the directory.' Below this are several form fields: 'Submitted By' with a search box and 'Clear' button; 'Requestee' with a search box and 'Clear' button; 'Start Date' with a date input field containing '8/6/2009' and a calendar icon; 'End Date' with a date input field containing '9/6/2009' and a calendar icon; and 'Format' with a dropdown menu set to 'PDF'. At the bottom are 'OK' and 'Cancel' buttons.

Figure 4-66 New user creation report parameters

The resulting report shows the time frame, for example, start date and end date, as shown in Figure 4-68 on page 217.

Tivoli Identity Manager

User Report

Report Criteria

Date Printed	17/04/2008
Time Printed	15:12
Time Zone	GMT-05:00
Data validity (as of)	11/04/2008 18:19
Report Generated By	itim manager
Total Entries Processed	89
User Input	Submitted By Like Any
User Input	Requestee Like Any
User Input	Start Date Greater Than 17/03/2008 00:00
User Input	End Date Less Than 17/04/2008 23:59

Request Type	Submitted By	Requested For	Subject	Status	Time Started	Time Completed	Last Access Time
Service Provision Process	System Administrator	Leigh Assauw	lassauw	Aborted	15/04/2008 16:57	15/04/2008 16:58	15/04/2008 16:57
Service Provision Process	System Administrator	Rob Hoffman	rhoffman	Completed	15/04/2008 12:37	15/04/2008 12:37	15/04/2008 12:37
Service Provision Process	System Administrator	Rob Hoffman	rhoffman	Completed	16/04/2008 17:31	16/04/2008 17:31	16/04/2008 17:31
Service Provision Process	System Administrator	Test User0001	0001	Completed	11/04/2008 17:57	11/04/2008 17:57	11/04/2008 17:57
Change Account Process	System Administrator	Test User0001	0001	Completed	14/04/2008 21:26	14/04/2008 21:26	14/04/2008 21:26

© 1999-2007 IBM. All Rights Reserved. Tivoli Identity Manager 6.0 Build 1484 1 of 8

Figure 4-67 New user report output

Report data synchronization

The Tivoli Identity Manager Report feature requires you to synchronize data and ACI information between the directory server and the Tivoli Identity Manager database tables, which are used to generate reports. Report synchronization is also required before Tivoli Common Reporting may be used.

Report data synchronization can be performed by:

- ▶ Scheduling data synchronization from the Tivoli Identity Manager GUI Reports Data Synchronization window
- ▶ Running the incremental data synchronizer

The incremental data synchronizer is a separately installed utility. There are some advantages to running the incremental data synchronizer rather than the regular data synchronization because:

- ▶ The incremental data synchronizer uses the changelog mechanism provided by directory servers.
- ▶ It is faster, as it processes only the changes since the last synchronization.

The incremental data synchronizer is not installed during the regular Tivoli Identity Manager installation. It must be installed separately.

4.6 Common customization

Tivoli Identity Manager can be customized in a number of ways. This section describes some of the more common types of customization, their uses, and the skills necessary to create them.

More complete descriptions of these, and other types of customization, can be found in the extensions directory of an Tivoli Identity Manager installation and in the IBM Tivoli Identity Manager Information Center Version 5.1 at the following location:

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

4.6.1 JavaScript extensions

JavaScript extensions alter the JavaScript environments that are used in the Tivoli Identity Manager server. Tivoli Identity Manager administrators can customize the behavior of the Tivoli Identity Manager server using JavaScript. Administrators can use scripts to calculate default values for account attributes, alter the content of e-mail messages, control the processing of workflows, and do many other actions. These scripts run as a part of the Tivoli Identity Manager server, and they should not be confused with client-side JavaScript that is embedded in Web pages to be run by Web browsers.

Tivoli Identity Manager provides a set of JavaScript environments. Each of these JavaScript environments provides predefined variables and functions that can be used by any scripts run in that environment. These predefined objects are created by JavaScript extensions. For example, all JavaScript run during the evaluation of a provisioning policy has a predefined variable named `subject`. This variable contains a reference to the owner of the account being evaluated with respect to the provisioning policy. This allows data about the owner to be used when calculating default or mandatory attribute values for their accounts.

Tivoli Identity Manager comes with a set of JavaScript extensions that create its default JavaScript environments. You can create your own extensions to add additional predefined variables and functions to any of the JavaScript environments.

Common reasons for creating custom JavaScript extensions are to:

- ▶ Allow scripts to access an external database or some other data source.
- ▶ Encapsulate a common calculation so that it can be used by multiple scripts.
- ▶ Allow scripts to use complex Java data types such as sets and maps.
- ▶ Allow scripts to make use of parts of the Tivoli Identity Manager Java API that are not exposed through an existing JavaScript extension.

Creating a JavaScript extension requires these skills:

- ▶ Java programming
- ▶ Enough knowledge of WebSphere Application Server administration to add your code to the Tivoli Identity Manager application
- ▶ Familiarity with JavaScript objects and the JavaScript namespace

Information about creating JavaScript extensions and some sample implementations can be found in the `extensions/5.1/doc/javascript` and `extensions/5.1/examples/javascript` directories of your Tivoli Identity Manager installation.

4.6.2 Application clients

An application client is a Java program that connects to the Tivoli Identity Manager server using a client-server model. The application client must authenticate to the Tivoli Identity Manager server as an existing Tivoli Identity Manager user. An application client can perform most of the functions that would be available to that user when using the Tivoli Identity Manager Web user interface. An application client may be either a stand-alone program or part of a Web-based application.

Common reasons to create an application client include:

- ▶ Creating an alternative self-service user interface for Tivoli Identity Manager
- ▶ Creating a self-registration application
- ▶ Creating a command-line interface for Tivoli Identity Manager

Creating an application client requires these skills:

- ▶ Java programming
- ▶ Familiarity with the Java2 and WebSphere Application Server security models

Information about creating application clients and some sample implementations can be found in the `extensions/5.1/doc/applications` and `extensions/5.1/examples/apps` directories of your Tivoli Identity Manager installation.

4.6.3 Workflow definitions

Tivoli Identity Manager uses workflows to define the basic operations that can be performed on persons and accounts. These operations include tasks like creation, deletion, modification, transfer, and password change. All of these workflows can be customized to redefine how these basic operations are performed.

Common reasons for customizing workflow definitions are:

- ▶ Adding approval steps to operations
- ▶ Automatically modifying an object attribute when the object undergoes a state change
- ▶ Propagating a change on one object to other related objects

Customizing workflow definitions requires JavaScript programming skills.

More information about workflow design and customization can be found in the IBM Tivoli Identity Manager Information Center Version 5.1, available at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_wkflo_planning.html

4.6.4 Workflow application extensions

Workflow application extensions define basic atomic services provided by Tivoli Identity Manager. These building blocks are the components with which operational workflows are built. Tivoli Identity Manager has system-defined application extensions that allow you to incorporate core Tivoli Identity Manager functionality into various Tivoli Identity Manager workflows.

You can also create your own application extensions. Each application extension is a method of a Java class. Your extension can define what inputs it expects to receive from the workflow and what outputs it will return.

Common reasons for creating workflow application extensions are to:

- ▶ Allow workflows to call custom Java code
- ▶ Encapsulate a common process so that it can be used by multiple workflows

Creating a workflow application extension requires these skills:

- ▶ Java programming
- ▶ Familiarity with the Tivoli Identity Manager data services Java API
- ▶ Enough knowledge of WebSphere Application Server administration to add your code to the Tivoli Identity Manager application

Information about creating workflow application extensions and some sample implementations can be found in the `extensions/5.1/doc/workflow` and `extensions/5.1/examples/workflow` directories of your Tivoli Identity Manager installation.

4.6.5 Password rules

Tivoli Identity Manager allows you to define custom password rules that may then be used in Tivoli Identity Manager password policies. You can also create a custom password generator to replace the Tivoli Identity Manager random password generator. You may need a custom password generator in order to guarantee that generated passwords comply with your custom password rules.

Creating custom password rules or a custom password generator requires these skills:

- ▶ Java programming
- ▶ Enough knowledge of WebSphere Application Server administration to add your code to the Tivoli Identity Manager application

Information about creating password rules and generators and a sample implementation can be found in the `extensions/5.1/doc/password` and `extensions/5.1/examples/password_rules` directories of your Tivoli Identity Manager installation.

4.6.6 Data services API

The data services API provides an abstraction layer above the Tivoli Identity Manager data stores. This API should be used when your customization must access the Tivoli Identity Manager directory or database. Using this API insulates your code from the actual representations of Tivoli Identity Manager data in the data stores, and makes your code more portable between Tivoli Identity Manager releases.

The data services API is not used by itself to create an Tivoli Identity Manager customization. Instead, it is used when you are creating some other type of customization that requires access to one of the Tivoli Identity Manager data stores. For example, you may want to create a JavaScript extension that searches for persons using a certain container in the Tivoli Identity Manager organization tree as the search base. Your JavaScript extension can do this using the data services API.

Using the data services API requires these skills:

- ▶ Java programming
- ▶ Enough knowledge of WebSphere Application Server administration to add your code to the Tivoli Identity Manager application

Information about the data services API and some sample code can be found in the `extensions/5.1/doc/dataservices` and `extensions/5.1/examples/dataservices` directories of your Tivoli Identity Manager installation. Note that the example code uses the data services API in a standalone program. This is not supported. It is done only to keep the example code as simple as possible. The only supported use of the data services API is in code that runs as a part of the Tivoli Identity Manager server.

4.6.7 Identity feeds

Identity feeds use data from an external data source to create, modify, and delete person records in Tivoli Identity Manager. Identity feeds are usually implemented using Directory Integrator. Directory Integrator provides simple and flexible ways to define how data should flow between data sources. It provides connectors that can read file-based data in a number of standard formats, such as comma-separated value, Extensible Markup Language (XML), and LDAP Data Interchange Format (LDIF). It also provides connectors that can extract data from structured data sources such as directories and databases.

Common reasons for using an identity feed are to:

- ▶ Keep Tivoli Identity Manager's person records synchronized with a human resources database
- ▶ Perform a mass update of person records in Tivoli Identity Manager

Creating an identity feed requires these skills:

- ▶ Familiarity with IBM Tivoli Directory Integrator
- ▶ Basic JavaScript programming

Information about creating identity feeds and sample implementations can be found in the `extensions/5.1/doc/idi_integration` and

extensions/5.1/examples/idi_integration directories of your Tivoli Identity Manager installation.

Information about IBM Tivoli Directory Integrator can be found at:

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

4.6.8 Custom person classes

Tivoli Identity Manager stores person and business partner person records as objects in the Identity Manager directory server. By default, person records are stored using the standard LDAP *inetOrgPerson* object class, while business partner person records are stored using the *organizationalPerson* object class.

You have the option of defining your own object classes to be used as custom person or business partner person types. You can define as many custom types as you like. The only restrictions are that your custom person classes must use *inetOrgPerson* as their super class, and your custom business partner person classes must use *organizationalPerson* as their super class.

Common reasons for creating custom person classes are:

- ▶ You must manage attributes that are not available on the default object class.
- ▶ You must apply different security rules to operations on different types of people.
- ▶ You must use different form definitions for different types of people.
- ▶ You must define custom operations on different types of people.

Creating a custom person or business partner person type requires the ability to define a new object class in Identity Manager's directory server.

You can find more information about creating custom person types in the IBM Tivoli Identity Manager Information Center Version 5.1, at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_entity_oview.html

4.6.9 Custom adapters

Custom adapters are used to manage accounts on platforms for which no predefined adapter is available. The preferred method for creating custom adapters is to use IBM Tivoli Directory Integrator. Directory Integrator provides connectors for many platforms. You can also create your own connectors using either Java or JavaScript.

Creating a custom adapter using Directory Integrator requires these skills:

- ▶ Familiarity with IBM Tivoli Directory Integrator
- ▶ Basic JavaScript programming
- ▶ Basic XML knowledge

Information about creating custom adapters and some sample implementations can be found in the `extensions/5.1/doc/idi_integration` and `extensions/5.1/examples/SampleLdapAdapter` directories of your Tivoli Identity Manager installation.

Information about IBM Tivoli Directory Integrator can be found in the Tivoli Directory Integrator Version 6.1.1 Information Center at the following location:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDI.doc_6.1.1/welcome.htm

If you want to use IBM Tivoli Directory Integrator 7.0 instead, more information can be found in the Tivoli Directory Integrator Version 7.0 Information Center at the following location:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDI.doc_7.0/welcome.htm

4.6.10 User interface

Aside from view configuration, Tivoli Identity Manager interfaces allow for some customization of their appearance and content.

Administrative console

The Tivoli Identity Manager administrative console user interface is customizable, although to a lesser extent than the self-care console. The header, footer, home page, and help redirection links of the administrative console interface are customizable.

You can customize the administrative console 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
 - Views
- ▶ Modifiable files
 - Properties files
 - Image files

Self-care interface

The Tivoli Identity Manager self-service user interface is highly customizable. Most graphical aspects of the self-care console can be modified.

The self-service interface can be customized 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
 - Views
- ▶ Modifiable files
 - Properties files
 - Cascading style sheet (CSS) files
 - A subset of Java server pages (JSP) files
 - Image files

More information about Tivoli Identity Manager interface customization can be found in the IBM Tivoli Identity Manager Information Center Version 5.1, at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_cfgsys_interface.html

4.6.11 Subforms

Subforms are custom windows that you add to the Tivoli Identity Manager Web user interface. Subforms are typically used to manage person or account attributes having values that cannot be easily displayed or edited with one of the standard Tivoli Identity Manager form control types. Attributes that contain XML data or multiple delimited fields are good candidates to be managed with a subform.

Subforms are displayed in a child window of the main Tivoli Identity Manager Web interface. The HTML content of the subform's window can be provided by either a servlet or a JSP. Your servlet or JSP is also responsible for parsing the attribute value to be displayed, and for formatting the new attribute value to be saved when the user has finished working in the subform.

Creating a subform requires these skills:

- ▶ Web page design
- ▶ J2EE programming using either servlets or JSPs
- ▶ Familiarity with the Tivoli Identity Manager applications API

- ▶ Enough knowledge of WebSphere Application Server administration to add your code to the Tivoli Identity Manager application

Information about creating subforms and a sample implementation can be found in the `extensions/5.1/examples/subform` directories of your Tivoli Identity Manager installation.

For more information about how to implement subforms consult the IBM Tivoli Identity Manager Information Center Version 5.1, at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/ref/ref_ic_subform.html

4.7 Adapter connectivity

All communication between the Tivoli Identity Manager server and its adapters is managed by the remote services module, as described in 3.1.2, “Service layer” on page 88. The remote services module communicates with adapters using objects called *service providers*.

Tivoli Identity Manager has three service provider types. You can also create your own custom service providers, but this is not a common customization. The remote services module determines which service provider to use when communicating with a particular adapter by looking first at the Tivoli Identity Manager service definition that maps to that adapter. The remote services module can then look up the service profile for that service. The service profile defines which service provider to use when communicating with any adapter mapped to a service with that profile.

Figure 4-68 shows how the remote services module processes an account request by determining which service provider to use, and then sending the request to that service provider.

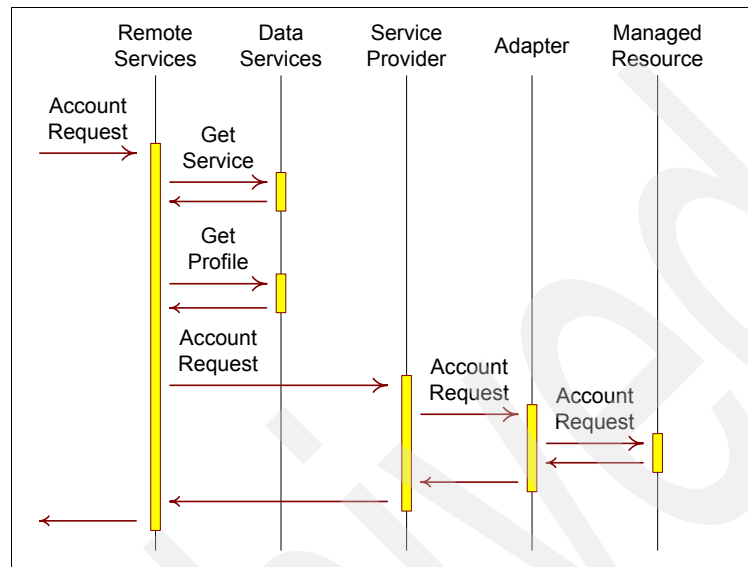


Figure 4-68 Remote services routes a request through a service provider

Each of the Tivoli Identity Manager service providers uses a different communications protocol to communicate with its adapters, as described in the following list:

- DSMLv2 (deprecated)** The DSMLv2 service provider was used to communicate with adapters implemented using IBM Tivoli Directory Integrator prior to using RMI. This service provider formatted requests as DSMLv2 messages, and sent these, via either HTTP or HTTPS, to Directory Integrator's DSMLv2 event handler. The event handler called the correct Directory Integrator AssemblyLine for the type of request. The AssemblyLine then used Directory Integrator connectors to carry out the request on the managed resource.

RMI

The Remote Method Invocation service provider is Tivoli Identity Manager's current mechanism to communicate with IBM Tivoli Directory Integrator. RMI technology uses Java calls over a distributed system, either in clear text or encrypted using SSL. This mechanism provides improved flexibility in communicating with IBM Tivoli Directory Integrator that is used to implement adapter account management requests as AssemblyLines. RMI also improves status code and return code handling.

RMI adapters differ from other adapters in that their service profile also includes a set of AssemblyLines. Each AssemblyLine defines how to perform a specific account management operation for that adapter's managed resource. For example, the LDAP RMI adapter features an AssemblyLine for add, modify, delete, search (for reconciliation), and test (for testing connectivity between Tivoli Identity Manager and the adapter). Change password, suspend, and restore are treated as modify operations.

DAML

The DAML service provider is used to communicate with most of the Tivoli Identity Manager adapters that are standalone programs instead of being implemented using Directory Integrator. This service provider formats requests as DAML messages and sends these, via either HTTP or HTTPS, to an adapter. The adapter then uses the managed resource's administrative APIs to carry out the request.

If you create your own custom service provider, then you are free to use whatever method you like to communicate with your adapters. You can even have your service provider manage resources directly without using an adapter (but this is not the preferred approach).

Managing resources directly with a custom service provider carries some risk. Your service provider runs as a part of the Tivoli Identity Manager server process. So any coding errors in your service provider (or client software used by your service provider) could cause the entire Tivoli Identity Manager server to fail. It is safer and preferred to create a custom adapter using Directory Integrator, and use the RMI service provider to communicate with this adapter. This protects the Tivoli Identity Manager server from things such as resource leaks and program hangs. If there is a problem in the adapter code, then the adapter can fail and be restarted without restarting the Tivoli Identity Manager server.

4.8 Software and hardware requirements

This section points the reader to online sources for obtaining information about the current software and hardware requirements for the Tivoli Identity Manager Version 5.1 and future releases.

4.8.1 Software requirements

Tivoli Identity Manager sits on top of a highly functional and sophisticated software stack that is constantly being updated by IBM developers and support engineers to:

- ▶ Improve performance
- ▶ Add new functions
- ▶ Automatically integrate functions between components in the software stack
- ▶ Add new components to the software stack from IBM and from other vendors

Figure 4-69 shows a sample Tivoli Identity Manager Version 5.1 software stack. As you can see, the number and the combination of products is extensive.

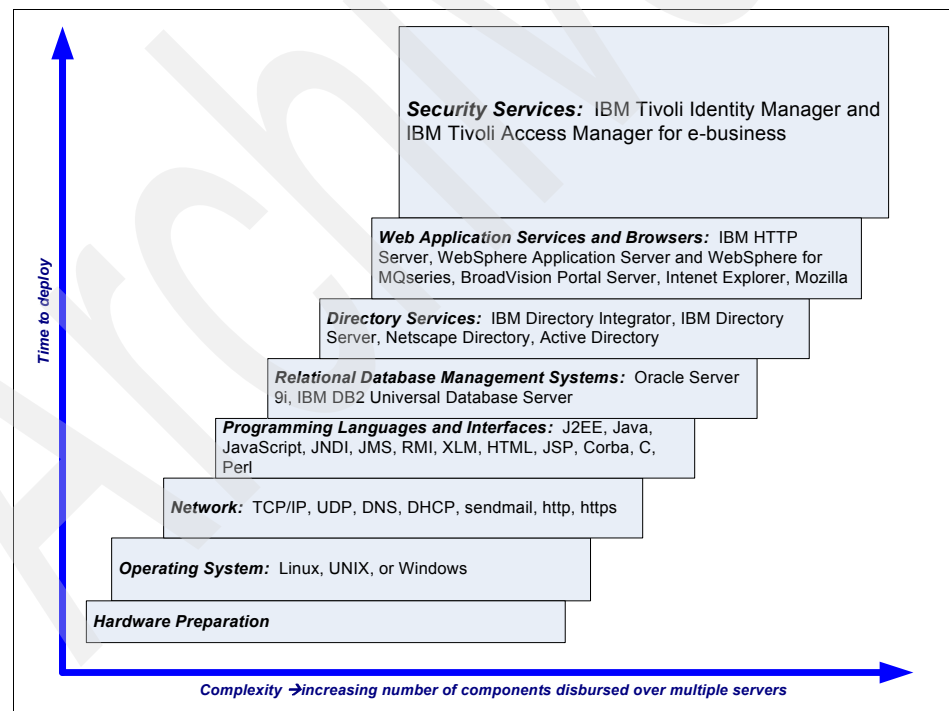


Figure 4-69 Sample software stack

The definitive source of information for managing the software stack is the *Release Information* in the IBM Tivoli Identity Manager Information Center Version 5.1 at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_release_oview.html

Frequent updates to this document are made. Download a copy of the Release Notes from the IBM Web site at least once a month to stay current.

The IBM Tivoli Software Information Center Web site is:

<http://publib.boulder.ibm.com/tividd/td/tdprodlist.html>

Additional sources of information include:

- ▶ The product documentation that is available at the IBM Tivoli Software Information Center Web site and IBM Redbooks publications at:

<http://www.ibm.com/redbooks>

- ▶ The IBM developerWorks® Security Management zone at:

<http://www.ibm.com/developerworks/tivoli/security/>

4.8.2 Hardware requirements

Hardware for an Tivoli Identity Manager Version deployment varies based on the business requirements, the functional requirements, and the availability requirements identified during the solution design process. In general, Tivoli Identity Manager solutions are packaged into non-clustered and clustered deployments.

Single server deployment

There are references in the product literature to a *single server* Tivoli Identity Manager deployment. In this configuration model, depicted in Figure 4-70, there is one WebSphere® Application Server instance running the Tivoli Identity Manager application. This configuration model may also be called a *non-clustered* deployment.

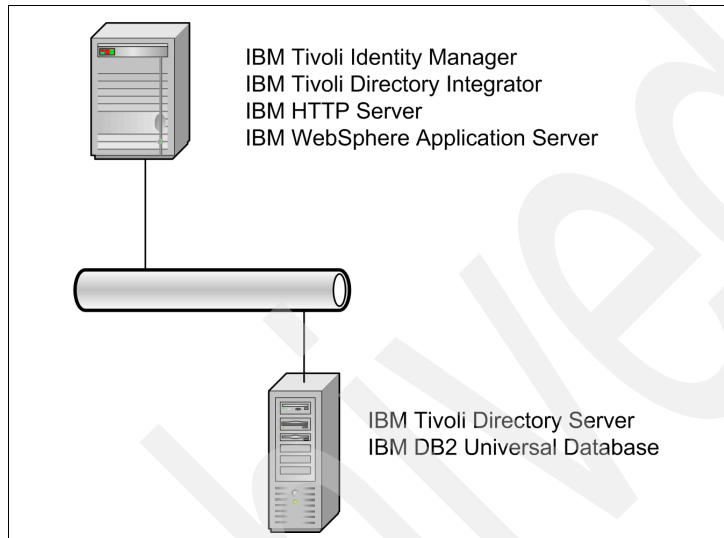


Figure 4-70 Sample single server configuration

All of the products in the Tivoli Identity Manager software stack may be placed on a single server provided that the server has sufficient CPU power, memory, and disk space to handle the workload. However, most deployments use two servers to improve the performance, where server 1 runs the WebSphere Application Server/Tivoli Identity Manager software and server 2 runs the directory server (LDAP) and the relational database (RDBMS) software.

This configuration model may be used for development, unit testing, and systems integration testing. It may also be used for small production deployments where redundancy for maintaining steady state operations may not be needed.

Cluster deployment

In this configuration model there may be two or more WebSphere Application Server instances each running separate Tivoli Identity Manager applications, under the control of the WebSphere Network Deployment Manager. There may also be separate servers for running the directory server (LDAP) and the relational database (RDBMS) software, in addition to separate servers for directory and database replication.

This configuration model may be used to scale Tivoli Identity Manager to handle large enterprise deployments and maintain steady state operations. More details about system-wide high-availability configurations can be found in 5.3, “High availability and failure recovery” on page 241.

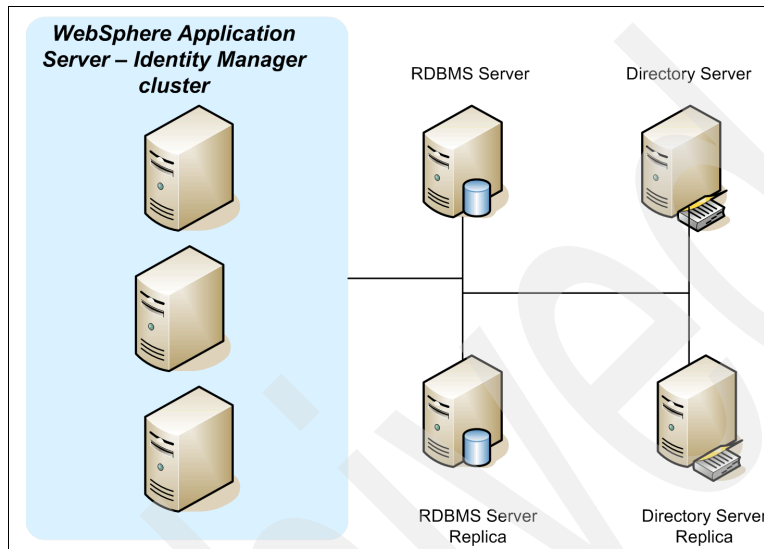


Figure 4-71 Sample cluster configuration

Note: If you chose a clustered deployment configuration model it may be advantageous to configure non-production Tivoli Identity Manager instances as clusters even though the cluster may only contain a single WebSphere Application Server/Tivoli Identity Manager instance. This increases productivity and knowledge transfers because you are using the same configuration model. This technique is especially useful when debugging.



Operational solution design

In this chapter we discuss the aspects of an Tivoli Identity Manager solution design not directly related to the business functional requirements. We address considerations to take into account when designing the non-functional and operational aspects of implementing and maintaining an Tivoli Identity Manager deployment.

Note: As continued widespread acceptance and recognition of it grows, many organizations are turning towards the use of the Information Technology Infrastructure Library (ITIL®), which was developed by the British Government's Central Computer and Telecommunications Agency (CCTA) as a basis for managing their service delivery and support needs. ITIL deals with areas such as:

- ▶ Service level management
- ▶ Cost management
- ▶ Contingency planning
- ▶ Capacity planning
- ▶ Availability management
- ▶ Configuration management
- ▶ Help desk
- ▶ Problem management
- ▶ Change management
- ▶ Software control and distribution

While this chapter is not intended to discuss or map Tivoli Identity Manager non-functional specifics to ITIL disciplines, nor is it explicitly stated in the various sections, you will find that most ITIL aspects that may apply to an Tivoli Identity Manager environment are covered within the topics discussed.

To learn more about ITIL visit:

<http://www.itil-officialsite.com/home/home.asp>

To avoid misunderstandings in this chapter, Table 5-1 illustrates user data and configuration items that are relevant specifically to Tivoli Identity Manager deployments. It also maps each item to those aspects of operational concepts that we discuss later on. Both the item list and the mapping are guidelines only. They suit common requirements. You might need to extend the table based on enhanced requirements in your company environment.

Table 5-1 User data and configuration items

Data category	Data item	Relevant at		
		Config. Mgmt.	Archive	Backup
Life cycle user data	Person		x	x
	Account		x	x
	Organizational unit		x	x
	Supporting data		x	x
	Audit and report data		x	x
Dynamic config data	Adoption policy	x		x
	Tivoli Identity Manager group	x		x
	Identity policy	x		x
	Life cycle operation	x		x
	Life cycle rule	x		x
	Organizational rule	x		x
	Password policy	x		x
	Provisioning policy	x		x
	Service	x		x
	Service selection policy	x		x
	Workflow design	x		x
	Custom person entity	x		x
	Adapter profile	x		x
	Report definition	x		x
Separation of duty policy	x		x	

Data category	Data item	Relevant at		
		Config. Mgmt.	Archive	Backup
Static config data	Java property	x		x
	Java extension	x		x
	Custom logo	x		x

5.1 Maintainability and configuration management

The ability to maintain and manage the configuration aspects of any project can be challenging. While there are general factors to consider that can be determined from project management concepts, there is usually the need to factor the specifics of the particular type of solution into the equation. This section outlines the high-level considerations that may need to be factored into an Tivoli Identity Manager deployment.

5.1.1 Version control

Software vendors, including IBM, release fixpacks periodically for all software products. Tivoli Identity Manager and the various major components of the Tivoli Identity Manager solution are no different. In the case of IBM, announcements are made when fixpacks are released, and there is the ability to subscribe to these through the IBM support Web site, which can be found here:

<http://www.ibm.com/software/sysmgmt/products/support>

The following Tivoli Identity Manager software components must be highlighted:

- ▶ Tivoli Identity Manager application
- ▶ Tivoli Identity Manager adapters
- ▶ IBM Tivoli Directory Integrator
- ▶ IBM WebSphere Application Server
- ▶ IBM DB2® UDB Enterprise Edition
- ▶ IBM Tivoli Directory Server

Note that Tivoli Identity Manager is supported on other software platforms for a subset of the components mentioned, but for the purposes of this discussion we list IBM components only.

There must be an evaluation and decision made by the project team to determine whether a component must be upgraded, whether it will be supported, the risks involved, and the benefits to be realized. As with all software, there must be a documented upgrade path specified for each component that should be tailored to suit the specific environment into which it is to be brought. A typical high-level approach involves moving the changes through development, test, and production once verification and regression testing is performed on the development and test environments. This must be done via the change management procedures relevant to the project. The levels of software versions should at least be kept consistent between the production and test environments. It is best practice to keep all environments at the same software levels.

Tivoli Identity Manager application configuration

It is not the goal of this discussion to address the general aspects of configuration management and version control for software development, project implementation, and change management in an operational environment. These issues are subjects in their own right and are addressed by many external sources dealing with systems development and implementation. Here we highlight version control and configuration management guidelines to some aspects specific to the Tivoli Identity Manager, such as:

- ▶ Static configuration data

Tivoli Identity Manager stores most of its application-specific configuration settings in Java properties files within its data directory. While backups may take care of ensuring that these files are not lost in the event of failure, it may be appropriate that they are also tracked using the version control procedures and tools of the project to track critical changes to Tivoli Identity Manager configuration. Examples include manually documenting the changes and leveraging the change control procedures and tools. Or use a version control system to track the different versions of the file by storing them.

- ▶ Custom Tivoli Identity Manager person objects

These are defined using both Lightweight Directory Access Protocol (LDAP) schema settings and LDAP data stored in Tivoli Identity Manager. If the definition of a type of person object changes, updates can happen to the LDAP data and the LDAP schema. Backups can assist with ensuring that this data is available in the event of a failure, but it may be appropriate that they are also tracked using the version control procedures and tools of the project. The required data for these changes are stored only in LDAP, so an example approach may be to extract the LDAP schema of the objects and attributes in question, in addition to the Tivoli Identity Manager LDAP entries corresponding to the object definition within Tivoli Identity Manager, as LDAP Data Interchange Format (LDIF) files before changes are made and use version control tools to track the different versions.

- ▶ Tivoli Identity Manager entities:

- Adoption policy
- Identity Manager group
- Identity policy
- Life cycle operation
- Life cycle rule
- Organizational rule
- Password policy
- Provisioning policy
- Service
- Service selection policy
- Separation of duty policy

- Recertification policy
- Workflow design
- Adapter profiles

These are all stored within the Tivoli Identity Manager LDAP, as they are application-specific pieces of data driven by business rules. At any point in time, Tivoli Identity Manager only keeps one single effective version of each entity object within the system. As these change over time, it may be decided that these should be version controlled and tracked against the business requirements at the time. The Tivoli Identity Manager import/export functionality can assist with extracting the definition of the relevant objects or the project may take a different approach such as directly extracting the data from the LDAP.

- ▶ Report definitions

These are stored within the Tivoli Identity Manager relational database. Each report definition can have a single effective copy stored at any point in time. Changes do not leave traces of older definitions of the particular report. If the report definitions are to be version controlled, the relevant database table rows should be extracted, and then have your version control procedures and tools manage this data. Refer to the *IBM Tivoli Identity Manager Database and Directory Schema Reference Version 5.1, SC27-2413*, for database schema details.

5.1.2 Multiple environments

As with any software implementation, an Tivoli Identity Manager deployment normally consists of multiple environments. Although there are inevitably variations of these, any environment can typically be classified under one of the following three categories:

- ▶ Development
- ▶ Test
- ▶ Production

Tivoli Identity Manager deployments are usually phased. That is, a structured and phased approach is taken for the rollout, thus adding to the complexity of the solution as a whole. For example, the initial phase may be called phase 1 and have a defined scope. This scope may consist of provisioning to a handful of platforms and have specific provisioning policies associated with these services. Phase 2 may expand on the scope and provision to more platforms and introduce new users into Tivoli Identity Manager to be managed. Subsequent phases will continue to increase the scope and size of the deployment.

In a stand-alone development environment where a system does not interact regularly with a multitude of heterogeneous external systems, the planning involved with maintaining environments and version control is relatively straightforward. The development environments can essentially be considered to be *fenced off* from other systems using *stub* components to model the interaction with external interfaces or systems. Some test environments also adopt this strategy. That is, there does not need to be a real external system available to interact with until the system integration testing phase of development, where there must be assurance that all the components involved do indeed integrate.

An identity management solution should not be viewed as a stand-alone solution. Due to its nature it interacts with many other systems in the corporate IT environment. The key concept to note here is that an identity management solution does not only read information from external systems, it writes to them and normally has administrative privileges on these systems. Tivoli Identity Manager is no different, and as a result will interact with external systems such as Windows, UNIX, enterprise resource planning (ERP) systems, databases, and many other applications, and will require administrative privileges on all of them. As a result, Tivoli Identity Manager needs connectivity into these systems from day one of development, as there is no easy way to *stub* test most of these systems to provide the functionality as required for development and testing of the solution. The interaction with multiple external systems increases the complexity of the phased approach and requires additional planning and consideration.

To illustrate this, consider the following scenario. An organization has set up an Tivoli Identity Manager solution to manage accounts on a few target systems (or managed resources), in this case Windows Active Directory, AIX, SAP, and RACF. They have distinct development, test, and production environments for Tivoli Identity Manager. Ideally, there should be those distinct environments for each target system, too. Most organizations do not have the luxury of being able to provide the infrastructure to do this, especially if mainframe systems are involved. This leads to situations where one target system is used by more than one Tivoli Identity Manager environment. Tivoli Identity Manager provides several means to divide a single target system logically into separate systems, such as:

- ▶ Account filtering based on LDAP search filters
- ▶ Defining distinct search bases on subtrees of account data
- ▶ Using ACIs to control access to distinct account data

Figure 5-1 displays a sample combination of available target system and Tivoli Identity Manager environments.

Available Environments	Development	Integration Test	Acceptance Test	Production
Target System				
Windows	✓		✓	✓
Active Directory	✓	✓	✓	✓
AIX		✓	✓	✓
SAP			✓	✓
RACF				✓ *Testuser
Scope of TIM Environments	Development		Test & Acceptance	Production

Figure 5-1 Tivoli Identity Manager deployments

There is no generic solution or methodology to solve this issue because of the fact that all managed systems with accounts are different and exhibit different behaviors. It requires careful planning with all the system owners to determine how to segregate the various environments physically or logically.

Another point to mention briefly is that in certain cases, a user data feed into Tivoli Identity Manager from a source (human resources system, for example) may be employed as part of the solution to perform the initial data load or to ensure that user data in Tivoli Identity Manager is kept up to date with the authoritative source. In these cases, the fact that production data may not be available in non-production environments due to various reasons, such as privacy and legislative controls, must be factored into the solution. For example, to ensure that these controls are met, it may be necessary to use only a subset of the data or to filter the data by modifying the sensitive pieces of information before using in non-production environments.

Note: It is a best practice and we strongly recommend that there is at least one non-production environment that mirrors the production environment exactly. This is not limited only to the Tivoli Identity Manager specific components, but also to the systems managed by Tivoli Identity Manager. For example, if Tivoli Identity Manager manages Windows Active Directory users in production, there must be a Windows Active Directory environment that is at the exact operating system and patch levels as the production Windows Active Directory, and this is to be used by the Tivoli Identity Manager test environment. This increases the level of confidence that promotion of changes into production will be successful and more importantly ensures that issues in production can be replicated in a non-production environment.

5.1.3 Migration between environments

Different types of information may or may not be moved between environments. There are various aspects of an Tivoli Identity Manager deployment that project teams may want to migrate between environments.

Identity Manager entities

This section covers all the business-related data objects specific to the management of users and accounts, for example, workflows, roles, and provisioning policies. Development activities on an Tivoli Identity Manager project are typically driven by the implementation of business processes and requirements that will affect this data and, as a result, the migration activities between environments.

The import/export function available within Tivoli Identity Manager is designed to ease the migration of Tivoli Identity Manager data objects between environments. Promoting an object such as a provisioning policy from test to production, for example, involves exporting the specific provisioning policy from the test environment and importing it into the production environment. The export of Tivoli Identity Manager objects within Tivoli Identity Manager also saves the export file into the Tivoli Identity Manager relational database with a time stamp associated with the file. This provides a simple way to maintain versions of Tivoli Identity Manager objects within the system and allow rollback to a previous version in the event that it is required.

Most Identity Manager entities can be migrated using the import/export function. Exceptions to this rule are:

- ▶ Tivoli Identity Manager organization tree structural: Organization tree data is live user data rather than configuration data. Thus, it should not be in the scope of a configuration management concept. It still must be exported from the Tivoli Identity Manager LDAP directory and imported into the directory of the environment to which it is being migrated. The reason behind this is that many Identity Manager entities are linked to the organization tree (to define their scope), and therefore cannot be migrated without them.

It is best practice to use LDIF to export and import organizational tree data or Tivoli Directory Integrator to synchronize it between two Tivoli Identity Manager instances.

- ▶ Person objects: These are not normally migrated between environments because it is very rare for production users to exist within the development or test environments in the exact same form in which they appear in production with the same sets of data. In certain cases, there are privacy and legislative reasons in place to prevent this from occurring. If cases exist where users must be migrated, it is a matter of exporting the directory data from the source environment and copying it into the destination environment. Again, the usage of Tivoli Directory Integrator AssemblyLines is the preferred way for doing that.
- ▶ Managed account objects: It does not make sense to migrate account objects within one Tivoli Identity Manager environment to another, as the managed resources will be different in most cases. That is, the production Tivoli Identity Manager system should not be managing the same set of accounts as the test Tivoli Identity Manager system, as this poses an inconsiderable amount of risk to the managed account. Doing this implies that a test system is managing production data, and that is not something many organizations want to do, for obvious reasons.
- ▶ Report definitions: These must be recreated or the relevant rows copied from the Tivoli Identity Manager relational database. Refer to the *IBM Tivoli Identity Manager Database and Directory Schema Reference Version 5.1*, SC27-2413, for database schema details.
- ▶ Audit data: This is not normally migrated between environments, as they are specific to the environment on which they exist. Each environment has its own set of audit data and it does not usually make sense to migrate audit data between the different development environments. These are stored within the relational database, and hence migrating is a matter for copying the relevant rows within the database. Refer to the *IBM Tivoli Identity Manager Database and Directory Schema Reference Version 5.1*, SC27-2413, for database schema details.

- ▶ Reconciliation schedules: These are stored within the database. Refer to the *IBM Tivoli Identity Manager Database and Directory Schema Reference Version 5.1*, SC27-2413, for database schema details.

Reconciliation schedules tend to change between environments, so there might not be the need to move them between environments. However, if maintaining them via the Identity Manager Web interface is not appropriate, you should use the Identity Manager API to manage reconciliation schedules. The API allows retrieving, adding, updating, and deleting reconciliation units for services. Look for `com.ibm.itim.apps.recon.ReconManager` in the API documentation or use the sample scripts that are shipped with Tivoli Identity Manager (for example, `addReconUnit.sh` and `removeReconUnit.sh`).

- ▶ LDAP schema: Certain deployments define custom LDAP objects to represent various pieces of business-specific data. The most common customization for this purpose is made to support specific attributes to be stored for person objects within Tivoli Identity Manager. In these cases, changes to the LDAP schema must be made. For deployments where this is the case, the LDAP schema changes must be migrated between environments at the LDAP level. That is, the project team should follow LDAP migration best practices for migrating schemas between LDAP servers.

Tivoli Identity Manager application configuration

Data and configuration information specific to an Tivoli Identity Manager deployment for the use of the Tivoli Identity Manager application is stored in properties files on the file system. This controls such things as logging levels, display settings, encryption settings, and so on. These do not affect the business logic configured into Tivoli Identity Manager, and hence failure to migrate these settings will not result in changes to the business processes and logic.

In most cases, these settings should not be fully migrated between environments, as they have various implications. For example, having excessive amounts of logging enabled may be desired for the development or test environments, but will cause a performance impact to the production system. Also, settings such as the LDAP or database server URLs should not be migrated between environments.

The safest way to migrate configuration settings is through the use of documented procedures that must be adhered to and are tied in with the project change control procedures.

Tivoli Identity Manager infrastructure configuration

This relates to the configuration settings in the various software components leveraged by the Tivoli Identity Manager application. These include:

- ▶ IBM WebSphere Application Server
- ▶ IBM DB2 UDB Enterprise Edition
- ▶ IBM Tivoli Directory Server

Each of these components is a software component in its own right, and there are best practice migration strategies for moving configuration settings for each between environments that are not within the scope of this discussion. In the context of an Tivoli Identity Manager deployment, however, it is extremely rare to migrate configuration settings of these components between environments due to the specific requirements of each. For example, production might need to be configured for performance and efficiency, development for ease of use and problem determination, and test somewhere in between. This implies that the infrastructure components should not be configured the same for each environment and should be treated as separate environments with no common components (even though it is known that there are different Tivoli Identity Manager deployments). The crucial thing to note here is that each environment should be tuned separately based on the requirements. Refer to the *IBM Tivoli Identity Manager Performance Tuning Guide Version 5.0 & Version 5.1*, SC23-6594, for more details.

5.1.4 Managing system accounts

It is worth mentioning here that there must be consideration given to managing system accounts, for example, the *root* account in UNIX or the *administrator* account in Windows. It is not necessarily a business requirement to manage these accounts, and as such they sometimes are not considered. Common strategies for managing system accounts within Tivoli Identity Manager are:

- ▶ Adopt system accounts.

This causes system accounts to be governed by Tivoli Identity Manager policies and have actions performed on them audited. There must be a decision made with regards to the rightful owner within Tivoli Identity Manager. There are two common options:

- System accounts can be owned by real people in the system.
- Functional people can be created within Tivoli Identity Manager to own the relevant system accounts (that is, person objects that do not define a real person, but rather act as a *placeholder* person object to own system accounts).

There are implications to the two approaches, the most obvious one being that if a real person owns system accounts, what happens to the accounts if the person is suspended or de-provisioned? There is no correct answer here. The course of action is specific to each organization and the relevant operation workflow can be modified to cater for the requirement.

- ▶ Do not reconcile into Tivoli Identity Manager.

This leaves the system accounts unknown to Tivoli Identity Manager. This is done via reconciliation filters. This means that system accounts are not subject to the policies defined within Tivoli Identity Manager. See 6.2.4, “Reconciliation” on page 286, for an example of this.

- ▶ Leave system accounts as orphan accounts within Tivoli Identity Manager.

This leaves system accounts known to Tivoli Identity Manager but not necessarily governed by the relevant policies. It also risks the accounts being de-provisioned or suspended if business rules mandate that orphan accounts are de-provisioned or suspended.

5.2 Archival and backup

In this section we detail the high-level considerations related to the archival and backup of an Tivoli Identity Manager deployment. The focus is on the critical software and data components required by the Tivoli Identity Manager application itself. The archival and backup of data on the target systems managed by Tivoli Identity Manager (for example, Windows Active Directory) is different for each type of managed resource and is not within the scope of this discussion.

First, let us define our common understanding of the *difference between backup and archival concepts*:

- ▶ In our view, a *backup concept* is designed to allow short-term disaster recovery of systems. Typical backup systems store images of active data on defined schedules and keep them for a limited time period, most often only a few days or weeks.
- ▶ An *archival concept* is designed to allow access to business information over a long period of time. Archives typically must be kept for a number of years. Often archives are maintained to meet with legislation and corporate governance practices like the Sarbanes-Oxley Act (SOX).

5.2.1 Archival

Archival of Identity Manager data may be required at various stages for varying business and operational needs. For example, increasing system performance and efficient usage of available system resources would be a common reason for data archiving. 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.

When designing an archiving solution some key questions are:

- ▶ Which data must be archived?

As indicated in Table 5-1 on page 225, not all of the Tivoli Identity Manager data requires archiving. In most cases it does not make sense to archive configuration data. This does not mean that it cannot be done, but is not common practice (backup of configuration data, however, is common and is discussed below). It is more common to have requirements to archive Tivoli Identity Manager live user data that is of no use for day-to-day operational and functional requirements, but must be retained for other reasons such as audit and compliance.

- ▶ How will the archived data be used?

As an example, if the requirement is to have archived Identity Manager audit data available on short notice in order to generate ad hoc reports of historical time frames, a different approach is needed than if the requirement is to be able to provide a printed list of all employees at a specific date in the past.

With this being said, we want to look at objects that are within the scope of an archival solution specific for Tivoli Identity Manager.

Person objects

Person details of people within the organization whose accounts are managed by Tivoli Identity Manager in most cases are maintained in human resources (HR) systems, which have their own archival strategy. Thus, there would be no need to archive those objects as Tivoli Identity Manager objects additionally. There are cases, however, where business requirements dictate that data related to a person cannot be deleted from the environment even when this is no longer required for day-by-day operations. They may be there only to satisfy business controls such as audit and compliance requirements. Having many records of this type within Tivoli Identity Manager can cause performance degradation due to the fact that Tivoli Identity Manager continues to take these records into account when performing various operations such as user-data-related searches or policy evaluation and enforcement. The implications of this must be considered and the relevant processes integrated with the operational environment.

There are various strategies that can be adopted, which vary based on each environment's requirements. An example approach is to archive the relevant objects into a separate *user archive* LDAP or an LDIF file and delete the live Tivoli Identity Manager LDAP objects. The deletion should be done via the Tivoli Identity Manager Web interface or the Tivoli Identity Manager APIs. If you intend to delete person objects directly at the LDAP server interface, you must make sure that corresponding objects such as accounts are be deleted or at least updated as well.

Managed account objects

Account objects in most cases already are covered by archival concepts of the corresponding target systems. However, similar to what has been said previously with person objects, business or audit rules might require archiving Identity Manager account objects. In this case we recommend using the same approach as described with person objects (that is, archive account objects in a separate LDAP instance or as a LDIF file).

Tivoli Identity Manager management objects

In rare occasions there is the requirement to not delete Tivoli Identity Manager objects in general, including management objects that govern the business rules represented within Tivoli Identity Manager. This can potentially result in requiring Tivoli Identity Manager to hold objects that are no longer required for operational or functional reasons (for example, a provisioning policy that is no longer used but must continue to be stored for audit purposes). While a good backup strategy covers the requirement to store this data, there may be additional requirements that mandate the data to be accessible on demand. There are various options for doing this. They can range from archiving the relevant data directly from the LDAP to an LDIF file to storing the data within another LDAP or persistent data store. It may also be feasible to use the import/export functionality within Tivoli Identity Manager for the objects that it supports, as exporting an object results in the storage of the relevant object and its dependencies within the Tivoli Identity Manager database as a JAR file. This may suffice in certain cases as the JAR file can be imported into a different Tivoli Identity Manager instance at a later stage or be used as the source of data for another software component that allows the viewing of the data. For example, IBM Tivoli Directory Integrator can be used to read data from the JAR file (or multiple JAR files) and used to produce the relevant output required.

Tivoli Identity Manager recycle bin

In IBM Tivoli Identity Manager Version 5.0, the recycle bin is disabled by default. If you have business or functional requirements that require you to enable this feature, all objects in Tivoli Identity Manager that are deleted get moved to the Tivoli Identity Manager recycle bin (“ou=recyclebin” subtree of the LDAP). There may be a business requirement to also archive the objects in the recycle bin before they are deleted. This can be archived in another LDAP or in an LDIF file. For further information about enabling and disabling the recycle bin, see the *IBM Tivoli Identity Manager Performance Tuning Guide Version 5.0 & Version 5.1*, SC23-6594.

Tivoli Identity Manager audit records

Audit records detailing all Tivoli Identity Manager operations are stored within the relational database. For further details, refer to *IBM Tivoli Identity Manager Database and Directory Schema Reference Version 5.1*, SC27-2413. These audit records can build up over time as Tivoli Identity Manager does not remove these. There are various reasons for this including the fact that audit records must be stored to be able to maintain the audit trail, and the ad hoc reports within Tivoli Identity Manager may require the historical audit information to be able to provide the person executing the report with information based on the audit trail.

As the audit records accumulate, the database audit tables become larger in size and hence there is a performance impact on the database. Subsequently, there is an impact on any Tivoli Identity Manager operation that interacts with the audit tables within the database. The *IBM Tivoli Identity Manager Performance Tuning Guide Version 5.0 & Version 5.1*, SC23-6594, details various steps to tune the Tivoli Identity Manager database, but this should also be combined with an archival strategy if required.

There are varying approaches that can be taken to archive the audit tables within the database. A decision must be made with regards to where the records must be archived and how to access them afterward. For example, they may be archived in the form of a database backup and written to disk or backup tapes. This depends on the audit requirements of the business. If the data that is archived is to be available when required, then it more than likely must be stored in an active database that is easily accessible rather than a backup file system where it must be restored to be viewed. The archived audit data can be stored in the Tivoli Identity Manager database or in a separate database instance. Note that the archival procedure should delete the relevant records from the audit database tables to allow for the performance gains to be realized. This typically involves database re-tuning by the database administrators to maximize the performance.

Important: The fact that the data is no longer within the audit database tables implies that Tivoli Identity Manager does not have direct access to the archived data. This includes standard Tivoli Identity Manager reports no longer being able to reflect the now archived data.

Examples of the various approaches that can be taken include (but are not limited to):

- ▶ Move older audit data periodically from the Tivoli Identity Manager audit tables to a separate archive table. Use database table views that combine the audit data in the Tivoli Identity Manager audit tables and the archived audit tables to consolidate all data into one logical location that can be used as a reference point for audit tracking and reporting.
- ▶ Set up table partitioning (DB2, ORACLE, Microsoft SQL Server 2005) for your audit database. This splits tables and indexes into smaller components and allows easier backup, restore, and merging of portions of your audit data.
- ▶ Move older audit data periodically from the Tivoli Identity Manager audit tables to an external source (for example, IBM Tivoli Data Warehouse) that is to be used as the reference point for all auditing and reporting activities.
- ▶ Move older audit data periodically from the Tivoli Identity Manager audit tables to a separate archive table. Use database queries or other software (for example, IBM Tivoli Directory Integrator) to consolidate the data from the various database tables to produce the required information.

5.2.2 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 includes the data within the Tivoli Identity Manager data stores. It also includes the relevant file systems or home directories on hard drives on which any of the Tivoli Identity Manager software components are installed.

File systems

The file systems or directories of the following Identity Manager software components must be backed-up as per the procedures of the environment in question:

- ▶ Operating-system-specific files of each machine that has an Tivoli Identity Manager software component installed.
- ▶ Home directory of each Tivoli Identity Manager application instance.

- ▶ Home directory (WAS_PROFILE_HOME) of each WebSphere Application Server¹ instance, and home directory of the Deployment Manager, if applicable, that hosts the Identity Manager application (itim.ear).
- ▶ Each LDAP instance. This does not include the LDAP data itself. It includes the home directory of the LDAP instance owner containing the instance configuration files and the directory of the LDAP server program files.
- ▶ Each relational database instance. This does not include the data stored within the databases. It includes the home directory of the database instance owner containing the instance configuration files and the directory of the database program files.
- ▶ The home directory of each Tivoli Identity Manager adapter interacting with a managed resource.
- ▶ The home directory of each Tivoli Identity Manager reverse password synchronization component.
- ▶ The home directory of any IBM Tivoli Directory Integrator components interacting with the Tivoli Identity Manager application.

Tivoli Identity Manager application data

The data stored within the following components must be backed up as per the data backup procedures of the component in question:

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

Note: Ensure that the restoration procedures from the backup data are tested. This is extremely critical, as the ability for this to be restored in a disaster recovery or critical failure situation must be assured.

5.3 High availability and failure recovery

High availability is a general term and can mean different things to different people. We define high availability as combining software with industry-standard hardware to minimize outages by quickly restoring essential services when a system, component, or application fails. This implies that the restoration of services is not instantaneous, but the down-time is negligible.

¹ More information about backing up and recovering the application serving environment can be found at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tadm_snrmain.html

There is also the notion of *fault tolerance*, which relies on specialized hardware to detect a hardware fault and instantaneously switch to a redundant hardware component. Although this is apparently seamless and offers non-stop service, a high premium is paid in both hardware cost and performance because the redundant components do no processing. More importantly, the fault tolerant model does not necessarily address software failures, by far the most common reason for outages.

Several operating systems provide support for running high-availability configured environments, for example, IBM AIX High Availability Cluster Multi-Processing (HACMP™), Linux Clusters, and Windows Server Cluster. There are benefits to using operating-system-based high-availability solutions and, depending on the requirements, such a solution may be desirable. They are generally designed to address both software and hardware failures of a single source, and in most cases do an adequate job. However, they do not always protect against cascade failures (multiple software or hardware failures), and most importantly require manual intervention in the event that there are issues with the data sources, for example, a corruption of data in one of the data stores. The costs and complexities involved with such an approach also tend to be higher and, as always, there should be a cost-benefit analysis performed to help determine whether this is the correct solution for a particular deployment.

In the context of this discussion, the high-availability aspects of the Tivoli Identity Manager software components are addressed. The concepts of a fault-tolerant hardware configuration and high-availability operating-system-based infrastructure should be considered and evaluated for each deployment, and the costs weighed up against the risks.

The software components relevant to an Tivoli Identity Manager deployment when considering a high-availability solution are:

- ▶ Application server
- ▶ Java Messaging Service
- ▶ Directory server
- ▶ Relational database
- ▶ Tivoli Identity Manager adapters
- ▶ Tivoli Identity Manager reverse password synchronization components

5.3.1 Application server

The application server runs the Tivoli Identity Manager application that performs all the business-related operations and provides the Web interface to users. There is only one scenario to consider when designing the application server component for high availability: Run the application server in its native clustered mode of operation.

The application server used by Tivoli Identity Manager is IBM WebSphere Application Server, which provides the ability to run as a WebSphere Application Server cluster. For further details refer to:

- ▶ *WebSphere Application Server V6 Scalability and Performance Handbook*, SG24-6392
<http://www.redbooks.ibm.com/redbooks/SG246392/wwhelp/wwhimpl/js/html/wwhelp.htm>
- ▶ *WebSphere Application Server V6 Planning and Design WebSphere Handbook Series*, SG24-6446
<http://www.redbooks.ibm.com/redbooks/SG246446/wwhelp/wwhimpl/java/html/wwhelp.htm>

Note: It is a best practice to leverage the functionality available in a security reverse proxy component such as Tivoli Access Manager WebSEAL to perform the authentication and authorization for users into Tivoli Identity Manager. WebSEAL also automatically performs the load-balancing and failover aspects in the event of an application server instance failure. This is especially useful if running separate instances of the Tivoli Identity Manager application on separate application server instances.

5.3.2 Java Messaging Service

The Java Messaging Service (JMS) is an essential component of typical J2EE applications. It is used by disparate components within an IT environment to communicate both synchronously and asynchronously and also as an interprocess communication mechanism between separate parts of the same J2EE application.

Tivoli Identity Manager makes intensive use of Java Messaging Service, especially its workflow engine, which controls the execution flow of almost all entity life cycle operations and requires this communication feature.

Before focusing on the Tivoli Identity Manager view on JMS, we want to describe basic WebSphere concepts and terminology. More details on WebSphere Java Messaging Services can be found in the IBM Redbooks publication *WebSphere Application Server V6.1: System Management and Configuration*, SG24-7304.

In WebSphere Application Server V6.1, the embedded messaging server (MQ) found in WebSphere Application Server V5.x was replaced by the default messaging provider, which is supported by the service integration bus (SIB), a new component of WebSphere Application Server V6.1. The SIB also benefits from the improved high availability manager capabilities.

Service integration bus (SIB) and message engines (MEs)

The service integration bus provides a managed communications framework that supports a variety of message distribution models, reliability options, and network topologies. The service integration bus acts as the default messaging provider for WebSphere Application Server V6.1. When clients (single WebSphere Application Server or WebSphere Application Server cluster) connect to a bus, they are connected to it through a message engine. Figure 5-2 shows a sample scenario.

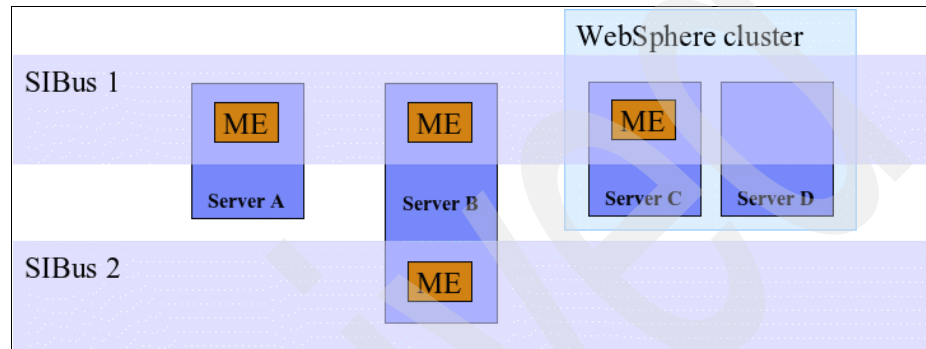


Figure 5-2 WebSphere Application Server service integration bus and message engines

Message engines hold one or more message queues. Every Message Engine has a data store that is used to store messages according to defined reliability levels. Messages that have been committed to the persistent data store are still available even after a queue has been stopped due to server shutdown or failure.

JMS delivery modes, reliability levels, and equality of service

Messages on the bus have a property called *reliability*, which defines how the messages will be delivered. There are five reliability levels available:

► **BEST_EFFORT_NONPERSISTENT**

Messages are never written to disk. Messages are discarded when a messaging engine is stopped, or in case of exceptions. Messages are thrown away if memory cache overruns.

This provides highest level of performance with the lowest degree of reliability.

► **EXPRESS_NONPERSISTENT**

Messages are written asynchronously to persistent storage if memory cache overruns, but are not kept over server restarts and failures. No acknowledgement that the ME has received the message.

- ▶ **RELIABLE_NONPERSISTENT**
Same as **EXPRESS_NONPERSISTENT**, except that we have a low-level acknowledgement message that the client code waits for before returning to the application with an OK or not OK response.
- ▶ **RELIABLE_PERSISTENT**
Messages are written asynchronously to persistent storage during normal processing and stay persistent over server restarts. If the server fails, messages might be lost if they are only held in the cache at the time of failure.
- ▶ **ASSURED_PERSISTENT**
Messages are never discarded. This has the highest degree of reliability where assured delivery is supported, but lowest level of performance.

WebSphere Application Server high availability manager

IBM WebSphere Application Server Network Deployment V6.1 comes with a new feature, the high availability manager, commonly called HAManager, to enhance the availability of singleton services in WebSphere as well as provide group services and group messaging capabilities to WebSphere internal components. These singleton services include:

- ▶ Transaction service (transaction log recovery)
- ▶ Messaging service (messaging engine restarting)

The HAManager runs as a service within each application server process (deployment manager, node agents, or application servers) that monitors the health of WebSphere clusters. In the event of a server failure, the HAManager fails over any singleton services that were running on the server that just failed. Examples of such services include the recovery of any in-flight transactions and restarting any messaging engines that were running. Figure 5-3 illustrates a messaging engine failover handled by the HAManager.

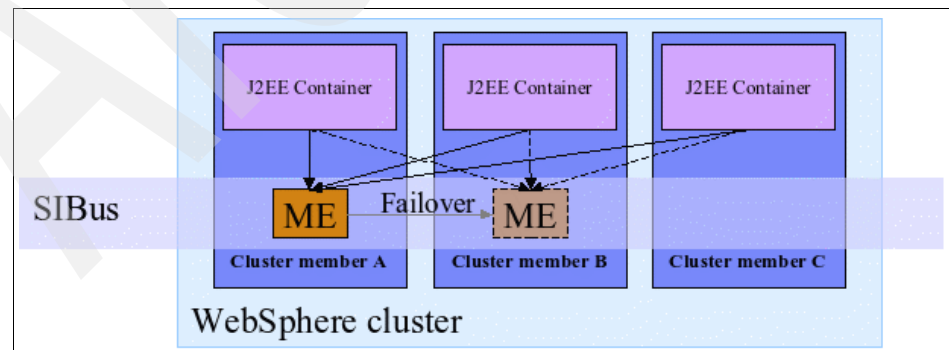


Figure 5-3 HAManager: Messaging engine failover

Tivoli Identity Manager usage of JMS

Now that we know about the JMS basics of IBM WebSphere Application Server we look at the Tivoli Identity Manager specific details of JMS.

Reliability levels of Tivoli Identity Manager JMS queues

For the sake of the highest possible reliability all but one queue are defined for ASSURED_PERSISTENT.

The one exception is the queue `itim_policy_simulation`, which is defined for RELIABLE_PERSISTENT. Since it is used for simulation of changes only, there is no risk of losing important process data if messages on this queue are lost.

Due to these default reliability levels in Tivoli Identity Manager messaging queues, no JMS message can get lost during graceful server shutdown or during server failure.

Local versus shared queues

At a high-level the JMS structure inside Tivoli Identity Manager uses two types of queues, local and shared.

Local queues

Local queues are used for operations that must be processed by a single Tivoli Identity Manager instance only. Those operations can be import/export processes or doing something like a large policy change, where we must read potentially millions of people off of the same search and take an action on each one.

Here is an example: Let us say that we are performing a provisioning policy change that affects 100 K people. This results in the need to run policy analysis on each one of those people, and potentially launch subprocesses to create or modify accounts. Tivoli Identity Manager starts an LDAP search for the affected people. It processes a chunk of those people in one transaction (let us say 1 K), sending messages with 50-person DNs onto the `itim_policy` queue. Once it has handled 1 K people, it sends a message to `itim_ps` telling it to continue the LDAP search in another transaction. The state of the LDAP search is stored in memory, so when the message arrives off of `itim_ps`, Tivoli Identity Manager can pick up where it left off, and because of how JMS transactions work, this will be handled in a new JMS message. If the message `itim_ps` did not arrive on the same server, Tivoli Identity Manager would lose its process state.

For the local queues, the messaging engines run in the WebSphere Application Server process itself. These MEs are statically assigned to individual cluster members, and do not have any automatic HA handling. This is intentional, since the messages on these queues must be processed on the server that sent them. Message storage is in the Tivoli Identity Manager database based on defined

JMS delivery modes. If a server fails during processing local queue messages, such as the one mentioned above, we end up with having those messages stored in the Tivoli Identity Manager database. No message will be lost, but since they are on local queues, they will not be picked up by any other cluster member. They will sit in the Tivoli Identity Manager database until either the failed server comes up again or a WebSphere Application Server administrator manually transfers the message queue to a healthy server.

There are seven local JMS queues used by Tivoli Identity Manager:

- ▶ itim_wf
- ▶ itim_ms
- ▶ itim_rs
- ▶ itim_rs_pending
- ▶ itim_adhocSync
- ▶ itim_ps
- ▶ itim_import_export

Shared queues

Shared queues are used for messages than can be picked up by any cluster member. For example, this could be any root workflow process within the Tivoli Identity Manager workflow engine or the subprocesses of large processes (such as handling each person's or account's subprocess in a role change).

For shared queues, the Tivoli Identity Manager installer sets up a clustered installation with a WebSphere high-availability policy that specifies that the shared queue messaging engine be run on one of the members of the Tivoli Identity Manager JMS cluster. So the shared queue ME starts up on one of the servers in that cluster and, if that server fails, the WebSphere HAManager transfers that ME (and any messages in the queues) to another server in the cluster automatically, based on the HA policies.

The three Tivoli Identity Manager shared JMS queues are:

- ▶ itim_wf_shared
- ▶ itim_policy
- ▶ itim_policy_simulation

Default Tivoli Identity Manager message queue setup

During the installation of Tivoli Identity Manager the setup program automatically configures the local and shared queues as appropriate to the underlying WebSphere infrastructure.

In a single-server installation, one messaging engine hosts all of the Identity Manager queues.

In a clustered installation with N application server nodes, there are N + 1 messaging engines. All *local* queues are hosted on N MEs hosts. Each ME is statically assigned to the individual application server nodes.

One ME hosts the *shared* queues. This ME runs in a dedicated WebSphere cluster, which by default is named *JMS Cluster*. There is a policy defined called *Default SIBus Policy* that applies to the shared queue ME. It is a *One-of-N* high-availability policy. Thus, it defines that one member of the JMS cluster actively hosts the queue with automatic failover to other cluster members in case of a server failure. The messaging engine starts on the first cluster member to become available. It is the standard high-availability manager in WebSphere 6.x associated with a service integration bus that handles the failover of this messaging engine and ensures that it runs in the cluster somewhere.

Figure 5-4 illustrates a typical Tivoli Identity Manager JMS configuration.

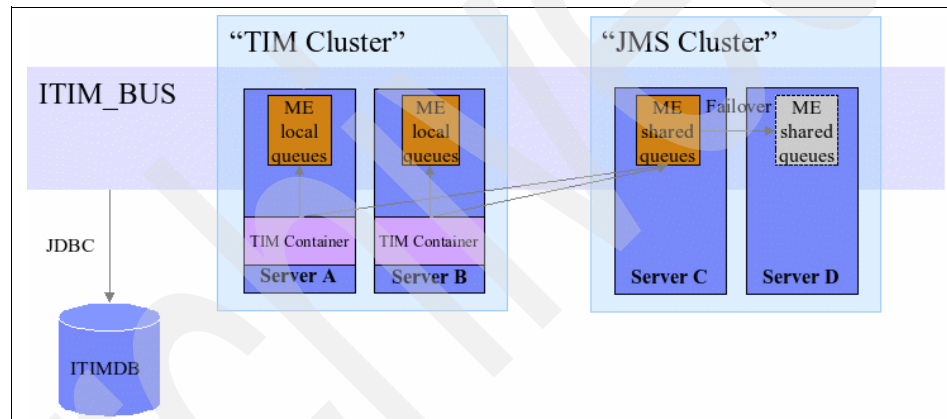


Figure 5-4 Typical Tivoli Identity Manager JMS setup

Conclusion

Tivoli Identity Manager leverages standard high-availability features of WebSphere 6.x to an optimal extent. Defining the maximum reliability level of `ASSURED_PERSISTENT` for all relevant messaging queues as well as using the high availability manager to control messaging engines according to service integration bus policies ensures reliable processing of Identity Manager entity operations without the need for additional high-availability techniques.

5.3.3 Directory server

Tivoli Identity Manager requires an LDAP server to store essential data such as users, accounts, policies, and so on. As a result, it is an extremely critical component. Most LDAP servers have some level of functionality to allow for a high-availability deployment through the use of data replication, as shown in Figure 5-5.

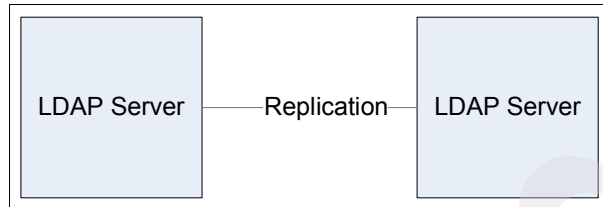


Figure 5-5 LDAP server replication

IBM Tivoli Directory Server, for example, allows for multiple LDAP servers to be configured with replication between them to ensure that data integrity is maintained. Each Tivoli Directory Server server can be configured as a read/write-enabled server or as a read-only replica.

Tivoli Identity Manager allows for configuration against a single logical LDAP. Note the use of the word *logical*. This means that it must refer to a Uniform Resource Identifier (URI) that will allow access to an LDAP. Given this, consider the following scenarios.

Note: Each of the scenarios presented below discusses high availability for LDAP servers generically. They are not necessarily specific to a particular LDAP product. For a complete list of LDAP servers supported by Tivoli Identity Manager, consult the latest release notes document.

Each of the scenarios presented below details two LDAP servers for illustrative purposes. The scenarios can easily be extrapolated to a topology that uses more than two LDAP servers.

Manual failover to secondary LDAP

Tivoli Identity Manager is usually configured to reference the physical location of the master LDAP server. Such a configuration looks similar to Figure 5-6.

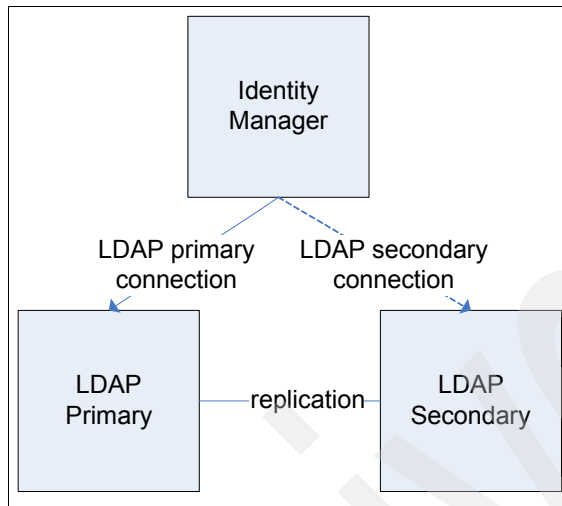


Figure 5-6 LDAP manual failover to replica

The issue is that in the event of the primary LDAP becoming unavailable, Tivoli Identity Manager must be configured manually to reference the secondary LDAP. This involves modifying the relevant Tivoli Identity Manager configuration file and restarting the Tivoli Identity Manager application.

For this scenario, it is imperative that both the primary and secondary LDAP have replication to each other enabled. Furthermore, the secondary LDAP should be configured as read/write, just like the primary. We do not run into problems with this configuration because the secondary LDAP is not used until after a manual switch from primary to secondary.

The main disadvantage of this approach as a whole is the manual intervention involved. The system is unavailable for a longer period of time due to the manual tasks involved and the requirement to restart the application. It may, however, be a less expensive option for certain deployments, and analysis of the costs against the benefits and risks may show this to be the most appropriate option.

Recovery from a failover situation

In regards to recovery from a failover situation:

1. The first step is to bring the failed primary LDAP back to a state where it is able to be used. Recall that in this type of configuration, both LDAP servers should be configured to accept and publish replication updates to each other. Thus, restarting the failed primary LDAP allows all the updates made to the secondary LDAP to be updated on the now functional primary.
2. Next there is the option to modify the Tivoli Identity Manager configuration to once again reference the primary LDAP and restart the application or to simply leave Tivoli Identity Manager referencing the secondary LDAP. This may be the approach taken to reduce the Tivoli Identity Manager application downtime. If both LDAP servers are able to handle the typical workload, it may not be necessary to revert back, and the secondary LDAP will become the new primary LDAP until another failover situation occurs.
3. There may be reasons specific to Tivoli Identity Manager deployments where it is preferable to switch the Tivoli Identity Manager configuration back to the original primary. For example, project documentation may have IP address or host name references, and not changing the configuration back causes the documentation to be inaccurate. These can obviously be factored into the change control procedures but again, this is depending on the specific deployment in question.

Automated failover to secondary LDAP

The use of a *smart* IP load balancer negates the need for Tivoli Identity Manager to be reconfigured in the event of an LDAP server failure. There are many load balancers on the market and the selection of a specific type is not discussed. The very high-level requirements for a load balancer to be used within the discussed environment are that it:

- ▶ Must be able to route application network traffic seamlessly and not modify data being routed
- ▶ Must be able to detect failure of a process accepting requests from the network
- ▶ Must be able to handle priority of potential destinations and route requests accordingly

In this case, Tivoli Identity Manager is configured to use the URI of the load balancer for its LDAP requests. The load balancer then forwards the request to the primary LDAP server. In the event that the primary LDAP is unavailable, the request is then forwarded to the secondary LDAP. The logic is handled at the load balancer and Tivoli Identity Manager does not need to be reconfigured in the event of an LDAP server being unavailable. An illustration of this can be seen in Figure 5-7.

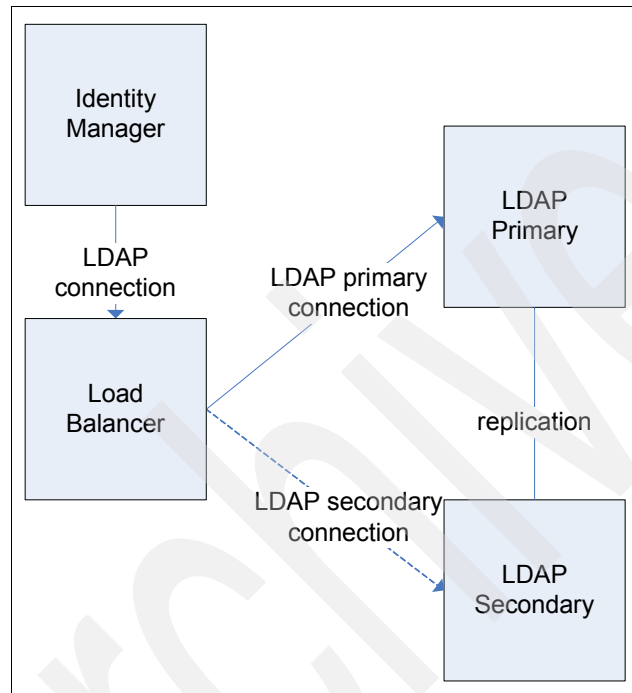


Figure 5-7 LDAP failover through a load balancer

As discussed in “Manual failover to secondary LDAP” on page 250, the secondary LDAP should be configured as a read/write LDAP and mutual replication between the primary LDAP and the secondary LDAP must be enabled.

The obvious disadvantage with this approach is the fact that the secondary LDAP is not being utilized until the primary LDAP fails. It is essentially functioning as a hot stand-by.

Note: It should be noted that because it is impossible to guarantee data consistency between directory replicas during all perceivable read/write scenarios, Tivoli Identity Manager *does not support load balancing* between itself and its directory servers, but only supports utilizing a single directory at any given time for its read/write operations.

Recovery

In regards to recovery from a failover situation:

1. The first step is to bring the failed primary LDAP back to a state where it is able to be used. Recall that in this type of configuration, both LDAP servers should be configured to accept and publish replication updates to each other. Thus, restarting the failed primary LDAP allows all the updates made to the secondary LDAP to be updated on the now functional primary.
2. If both LDAP servers are able to handle the typical workload, it may not be necessary to revert back, and the secondary LDAP will become the new primary LDAP until another failover situation occurs. This is the preferred configuration option.
3. There also is the option to modify the load balancer configuration to once again forward traffic to the primary LDAP. Care must be taken in this step due to the fact that updates may have been made to the secondary LDAP that must be replicated to the primary LDAP before new updates are made to the primary LDAP. If the priority on the load balancer maintains that the primary LDAP is the preferred option, then it starts to send LDAP update requests to the primary LDAP when it determines that it is available. If this occurs before the full list of replication updates are received and processed from the secondary LDAP, there may be potential issues in maintaining the synchronicity of the data between the LDAP servers due to the potentially large volume of changes being replicated to the primary LDAP.

There are various approaches that can be taken here. One approach is if the load balancer allows dynamic modification of its destination priorities, then this should be made before the primary LDAP is made available. This ensures that updates continue to be sent to the secondary LDAP until the primary LDAP is ready. Another approach is to ensure that the network does not allow the load balancer to connect to the primary LDAP (perhaps using a firewall) until it has received the replication updates from the secondary LDAP. This implies that the secondary LDAP is the only application in the network that can connect to the primary LDAP until it is ready to be used.

5.3.4 Relational database

Today most potential Relational Database Management Systems (RDBMS) come with built-in features to support high-availability and disaster recovery concepts. In addition to that, they can be tightly coupled with high availability operating system configuration options, systems management products (automation, virtualization), and storage solutions (for example, storage area networks).

The decision as to what products or features to use depends, as always, on the specific challenges of the environment, budget, complexity, time to implement, and degree of security required.

The major strategies for highly available RDBMS implementations are:

- ▶ Hardware redundancy

Hardware solutions usually have the highest performance and security by copying or replicating entire disks. Using a hardware solution reduces the granularity and ability to determine when and what must be replicated. These solutions are also usually much more expensive than software solutions (for example, RAID systems and Remote Storage Mirroring).

- ▶ Clustering solutions

The idea of clustering is to present to the users a single machine, when in fact the system has multiple nodes to serve the client applications. Clustering solutions can be broadly categorized in to two types:

- Operating system dependent, for example, IBM HACMP for AIX, Microsoft Windows Server 2003 Clustering Services, and Sun Cluster for Solaris
- Operating system independent, for example, Tivoli System Automation (TSA), and Veritas Cluster Server

- ▶ Replication

Replication basically describes methods for copying and distributing data from one database to one or more other databases. Database products might come with out-of-the-box replication features (like DB2 HADR, SQL, or Q replication) or you could use external tools to implement replication topologies.

Taking IBM DB2 UDB Version 9.1 as an example and using a combined approach leveraging operating system high-availability features and additional IBM products, the solution design team may choose to deploy DB2 as follows:

- ▶ Operating system cluster with DB2 active/standby configuration

In the event of the active DB2 instance failing, the operating system clustering software starts the same instance on another node in the operating system cluster. This requires that all nodes in the operating system cluster have access to the same shared disk. While relatively simple in terms of DB2 clustering, this introduces delays during failover while the new processes are started and any in-flight transactions are rolled back. The database is accessed through the cluster address, so that no change in the Tivoli Identity Manager database configuration is required during failover.

- ▶ DB2 mutual takeover multiple partition configuration

All nodes in the database cluster operate in parallel. The database is partitioned so that if any server in the cluster fails, its partitions are failed over to the remaining nodes in the cluster. As with other strategies, there are various considerations in using this approach. The configuration still requires time for the failed-over partitions to be recovered, although as each partition has less than the entire volume of data it is generally faster than an active/standby configuration. There must be database analysis performed to determine an appropriate database schema required for constructing a partitioned version of the Tivoli Identity Manager database. This approach also requires that all servers have access to the file systems containing the database and transaction logs. The database is accessed through the cluster address, so that no change in the Tivoli Identity Manager database configuration is required during failover.

- ▶ DB2 High Availability Disaster Recovery (HADR) in combination with Tivoli Systems Automation (TSA)

This solution uses the DB2 HADR feature to replicate data from a master database to a standby database. Note that only one server can be active and read and written to by a client. The secondary stand-by servers cannot participate in reads.

If the primary active server fails then the passive standby server can take over the role as primary server. Since a HADR primary database does not automatically switch over to its standby database in the event of failure, to achieve automatic monitoring and failover, Tivoli Systems Automation can be used with DB2. Tivoli Systems Automation monitors the HADR pair for primary database failure and issues appropriate takeover commands on the standby database in the event of a primary database failure.

In case of a failover, the Tivoli Identity Manager DB2 client is automatically rerouted to the secondary failover server. This requires the DB2 feature *automatic client reroute* (ACR) to be activated. The DB2 client takes care of the failover. Hence, there is no need for manual intervention in the Tivoli Identity Manager database configuration during failover.

Further details on how to implement this approach are available within the whitepaper “Automating DB2 HADR Failover on Linux using Tivoli System Automation for Multiplatforms,” available at:

ftp://ftp.software.ibm.com/software/data/pubs/papers/hadr_tsa.pdf

Note: The DB2 options outlined are intended as examples. These are not the exhaustive high availability strategies available with DB2. For specific details and other options, refer to the DB2 9.1 product documentation.

5.3.5 Tivoli Identity Manager adapters

Tivoli Identity Manager manages accounts on managed resources through the use of adapters. This section discusses adapters in general. Note that there are different adapters for each distinct type of managed resource, but the concepts discussed can be generally applied. For example, the concepts apply to the Windows adapter as well as the Linux adapter, and so on.

Account operations issued by Tivoli Identity Manager are executed by the relevant adapter for the type of managed resource on which the accounts reside. This includes provisioning, password management, and reconciliation operations. It can be argued that account operations are not mission critical and hence do not need to have high availability requirements factored in for a solution design. In many cases, this may be true. As with many things, however, there are exceptions to the rule. Each deployment has specific requirements and it may be decided that certain operations must be highly available. For example, there may be cases where password resets, account suspensions, and account de-provisioning are deemed to be critical operations and must be highly available.

There are two aspects to consider when dealing with high availability for Tivoli Identity Manager interactions with its adapters and subsequently the adapter interactions with the managed resource hosting the accounts being managed, as illustrated in Figure 5-8 on page 257. The first is the adapter interactions with the managed resource, for example, the Tivoli Identity Manager Windows Active Directory adapter and its interactions with Windows Active Directory. The high availability aspects between these two components are not within the scope of this discussion, as each managed resource has different approaches to high availability and they can vary in completely different ways.

The focus of this discussion is on the Tivoli Identity Manager specific components required for ensuring that account operations are highly available.

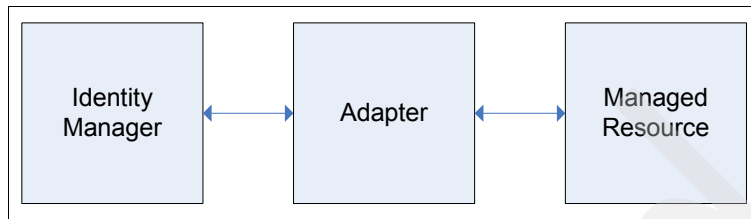


Figure 5-8 Tivoli Identity Manager interactions with managed resources

Tivoli Identity Manager, as per the approach taken with the LDAP and database, references its adapters via a URI. As previously mentioned, this is a *logical* location. Given this, consider the scenarios discussed in the following sections.

Note: The following scenarios consider the use of two adapters in each case. This can be extrapolated to cases where it is determined that there is a need for more than two adapter instances.

The scenarios presented below are failover scenarios. We do not recommend considering using a load-balancing strategy for the adapters. This can cause issues with operations such as reconciliations, which rely on using a dedicated adapter instance.

Manual failover to secondary adapter

This scenario involves using a secondary adapter to be available for use in the event of the primary adapter being unavailable, as shown in Figure 5-9. This may be an option if it is deemed that a short (but not negligible) amount of downtime is acceptable. The URI reference to an adapter is not stored within a configuration file in Tivoli Identity Manager. It is stored as an attribute of the service definition. Because of this, a change in this attribute does not require a restart of the Tivoli Identity Manager application. All that must be done is for the value to be modified and saved within the service definition for it to take effect. The assumption being made here is that the secondary adapter is configured exactly the same way that the primary adapter is configured. If not, additional attributes must be modified within the service definition. This depends on the difference in configuration settings between the two adapter instances.

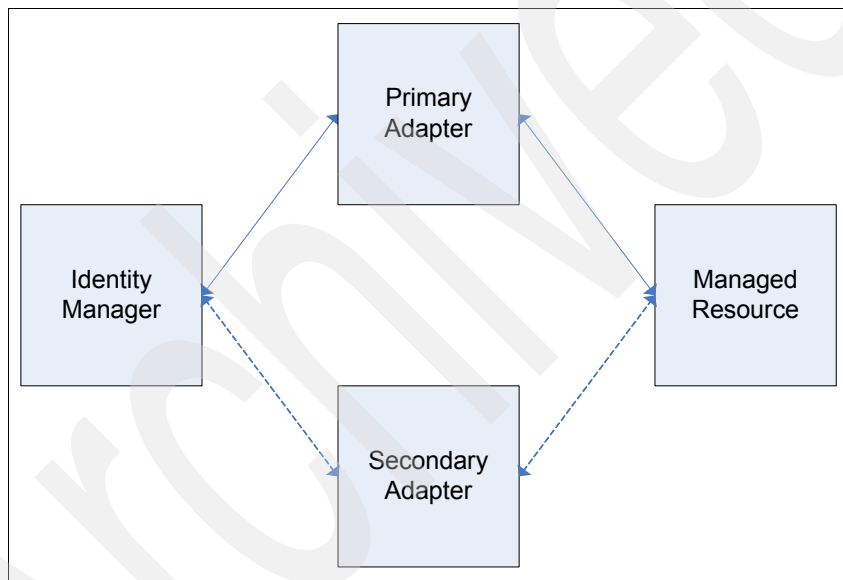


Figure 5-9 Manual failover to secondary adapter

If the primary adapter becomes unavailable, manual user intervention is required to make the change in the relevant service definition to reference the secondary adapter. This secondary adapter can be an active stand-by or it can be brought up to an active running state when it is required. An active stand-by secondary adapter results in a shorter amount of downtime but requires the system resources to be available for use by the secondary adapter. If the secondary adapter is to be brought to an active running state, it may take a little longer for availability to be achieved, but it does not consume system resources (other than disk space) while it is not required.

However, the resources consumed while in an active state but not being utilized by the Tivoli Identity Manager server are relatively insignificant. The approach taken depends on operational requirements of the organization (for example, the amount of system resources available or service level agreements that must be met).

Recovery from a failover situation

It may be acceptable to use the secondary adapter as the active adapter until such a time when it is unavailable and perform the steps mentioned to have Tivoli Identity Manager use the primary adapter again. If it is essential that Tivoli Identity Manager always use the primary adapter if possible, then a suitable approach is to wait for the next available change window to do so.

Automated failover to secondary adapter

This scenario relies on the secondary adapter being available to be used at all times. There is also a requirement to leverage the use of a suitable IP load balancer, as detailed in “Automated failover to secondary LDAP” on page 251. The Tivoli Identity Manager server is configured to reference the URI of the load balancer, which then routes requests to the relevant available adapter, as shown in Figure 5-10.

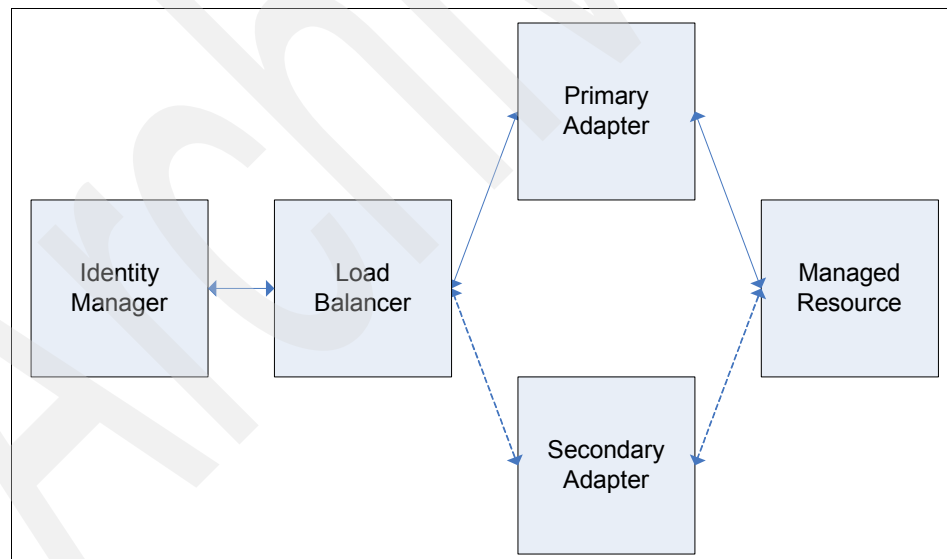


Figure 5-10 Automated failover to secondary adapter

An important point to note with this approach is the use of the word *failover*. The load balancer is configured to prioritize requests to the primary adapter. That is, it attempts to use the primary adapter at all times until such a time when it is unavailable. In the event of unavailability, the load balancer uses the secondary adapter. Note that the secondary adapter must be configured exactly the same way that the primary adapter is configured. If not, then additional scripting may be required to achieve automated failover to the secondary adapter.

Recovery from a failover situation

For recovery from a failover situation:

- ▶ Load balancer allows modification of priorities while it is running.

It should be modified to give priority to the secondary adapter while the primary adapter is being brought back to a functional state. The configuration can be left as is with all requests being routed to the secondary adapter until such a time when it is unavailable and the primary adapter is required, at which time it will automatically route requests to the primary adapter. If the desire is to use the primary adapter where possible, this means that the load balancer must be reconfigured to prioritize requests to be routed to the primary adapter. This should be done within the change control procedures of the environment and during a suitable change window.

- ▶ Load balancer does not allow modification of priorities while it is running.

The network connection between the load balancer and the primary adapter should be disabled (either directly at the network configuration level or via a device such as a firewall) while the primary adapter is brought back to an active functional state. The network connectivity can be restored within the change control procedures of the environment and during a suitable change window.

5.4 Monitoring

As with any best practice solution design, there must be a monitoring solution designed for an Tivoli Identity Manager deployment. A major driver for monitoring is to ensure that an organization's *service level agreements* are met to ensure business continuity. At a higher level, this can be traced back to the overall solution's quality of the service. An environment that is poorly monitored can result in having issues such as:

- ▶ Problems going unnoticed until user complaints are made. The business impact is low user satisfaction.

- ▶ Problems going unnoticed until the system fails. The business impacts are:
 - A team of specialists must react on short notice in order to de-escalate a critical situation rather than proactively addressing the outage.
 - An unplanned system outage puts SLA at risk.
- ▶ Systems are running with bad performance. The business impact is ineffective use of given hardware and network resources.

At the very least, a monitoring solution should be able to indicate to the relevant support personnel that a part of the system needs attention, whether it be visual monitoring of the component to ensure that it remains stable or to perform manual corrective actions to correct the issue. In certain cases, the monitoring solution may even perform automated correction of the failure to ensure continued availability.

In addition, monitoring solutions should provide functions for the tracking of system performance metrics. This allows for a quantitative, and in some cases visual, measurement of the indicators, factors, and patterns that constitute a *stable state* identity management solution. For example, a visual graph of system performance over a period of a week where the system does not exhibit any problems is a good reference point to use against continued monitoring of the system. It allows for variations in patterns to be analyzed, as they may indicate potential system issues.

There are many tools and products available that can be used to implement varying levels of monitoring. Note that each separate Identity Manager deployment has its own set of methodologies and processes governing the decisions and design specifics of their monitoring solution. As a result, the following discussion does not aim to provide a detailed set of specifications for monitoring an Tivoli Identity Manager environment. It aims to provide high-level considerations to use as a basis for designing a generic monitoring solution for an Tivoli Identity Manager solution.

An identity management solution such as Tivoli Identity Manager involves many components. This is not limited to the Tivoli Identity Manager specific components (for example, target systems where accounts are managed, such as Linux). While there is a separation of responsibilities between the interacting components, there are the integration aspects to consider. From an overall solution perspective, if a managed resource is unavailable, there is nothing Tivoli Identity Manager can do to remediate the situation. It simply cannot manage the accounts on the offending managed resource until it becomes available.

However, from the user point of view, it is the Identity Manager solution (as a black box) that is not functioning if she is trying to change her password at that moment. This suggests that an Tivoli Identity Manager deployment must be viewed as the complete identity management solution, especially in terms of solution availability and subsequently monitoring. Taking this approach lends itself to the various aspects of an Tivoli Identity Manager solution to consider when discussing monitoring.

5.4.1 Tivoli Identity Manager infrastructure

This section discusses the supporting hardware and software components required by the application, including, for example, the operating system, database, LDAP, and application server. There are prescribed methodologies with regards to monitoring the various infrastructure components important to Tivoli Identity Manager that are not discussed, as the approaches are not necessarily specific to an Tivoli Identity Manager deployment. For example, there are generic monitoring strategies for a relational database regardless of the applications using it involving, for example, connection pools, locks, table space, and so on. The same can be said about the other infrastructure components. All that must be mentioned in the context of Tivoli Identity Manager is that there must be a general monitoring strategy implemented for the following Tivoli Identity Manager infrastructure components:

- ▶ Operating systems
- ▶ Application servers
- ▶ Relational databases
- ▶ LDAPs
- ▶ Load balancers (if used as part of the high-availability solution design)

Note: This does not discount the fact that other services such as the network, firewalls, hardware devices, disks, backup procedures, and so on, need to be monitored as well. These are not directly related to a functional Tivoli Identity Manager environment, but failure in any of these organizational IT infrastructure components may cause Tivoli Identity Manager failures indirectly.

5.4.2 Tivoli Identity Manager application

A more appropriate discussion around monitoring in the context of this section relates to the Tivoli Identity Manager application itself. This includes monitoring application-specific events such as provisioning, password resets, and so on. In some cases, monitoring of the application may result in the realization that an infrastructure component is unavailable, but it would not be as a result of monitoring the infrastructure component directly.

For example, it may be determined that an Tivoli Identity Manager authentication attempt failure is due to the LDAP being unavailable. For cost reasons, the decision may be made to leverage the use of custom scripts written by the project team to allow for simple monitoring, but generally much of the application monitoring requires more sophisticated tools to allow for proactive actions to be taken and also to allow for additional features such as root cause identification and determination, historical tracking, reporting on usage, and transactional response time patterns for the various application functions being monitored.

Tivoli Identity Manager Web application

This contains the core Tivoli Identity Manager application components and resides on the application server. Tivoli Identity Manager is a J2EE application and as such can have monitoring applied to its J2EE components much the same way in which the infrastructure components are monitored. This can be tied in with the monitoring of the application server. More appropriate to this discussion, however, is to highlight the various Tivoli Identity Manager application operations that may potentially be monitored and approaches that may be taken. This includes (but is not limited to):

- ▶ **Web application availability:** Simple tests can be performed to confirm the ability to connect to the Tivoli Identity Manager application even as an unauthenticated user. Basic scripts to perform *ping* tests can be used or more sophisticated monitoring tools can be used to simulate user requests for a particular Web page (the first page that one sees once logged in, for example) and measure response times to proactively monitor the system.
- ▶ **Authentication:** Periodic monitoring of authentication into the Tivoli Identity Manager application ensures the availability of two facts:
 - The application is available.
 - The Tivoli Identity Manager LDAP is available.
 - The Tivoli Identity Manager database is available.

Authentication fails without the availability of the LDAP. Scripts can be used to combine operations that perform ping test of the application and also LDAP bind requests, but it is difficult to simulate actual Tivoli Identity Manager authentication events without the use of a monitoring tool or without leveraging a custom application that uses the Tivoli Identity Manager APIs to authenticate a user. A reduction in authentication response times may indicate additional stress on the LDAP or may indicate that the LDAP requires attention.

- ▶ Object search: Ensuring that searches can be performed via the relevant Tivoli Identity Manager Web application window ensures that the LDAP is available and that the performance of the LDAP is as expected. Performance degradation here may indicate that there are too many objects (person objects and accounts are the most likely objects to increase in number) in the system for the environment, and tuning of the LDAP may be required or an archival strategy may be appropriate to manage the unused objects and execute the documented archival procedures for the environment. Refer to “Archival” on page 237 for the discussion on archiving Tivoli Identity Manager objects.
- ▶ Pending operations: Any transactional operation within Tivoli Identity Manager generates a pending item in the application. A large number of pending operations usually indicates that a large bulk transaction is being processed (for example, a user data load or large provisioning policy change) or that there is an issue with the system. These are stored within the Tivoli Identity Manager relational database. Monitoring cyclic trends allows for system administrators to track a usage pattern for a stable environment and notice when something unexpected occurs. For example, it may be known that spikes in pending operation numbers are generated at certain times during the week due to a user data load, and this behavior can be verified by using the historical monitoring data. Any variance to the pattern can indicate a potential issue and proactive actions can be taken to correct the issue before a substantial degradation of the service is experienced.
- ▶ Person life cycle events: This includes create, change password, suspend, restore, modify, and delete. There should be periodic checks performed to ensure that people and their accounts are able to be created within the application. This may be done by *reserving* a test user in production that exists only while the monitoring tools are performing the verification that these operations are functional. For example, the sequence of events may be to have the monitoring tool create the person, create accounts, change their password, suspend the person, restore the person, modify an attribute of the person, and finally delete the person and the accompanying accounts. Part of the change password action may be to test that the reverse password synchronization components are functional on the managed resource. This is in addition to ensuring that password changes initiated via the Tivoli Identity Manager Web interface are functional. The monitoring tool can also track the transactional response times on each operation to ensure that the system is performing as expected. Any variance to the expectation may need to be investigated.

- ▶ Application audit events: Tivoli Identity Manager provides audit events for informational purposes. Error codes are also generated to indicate the type of issue occurring. Monitoring tools can be used to monitor these events for critical problems that arise and also potentially for security events. For example, users continually trying to authenticate to Tivoli Identity Manager and failing may indicate a *brute force password* attack.

Adapters

The monitoring of adapters is critical to continued Tivoli Identity Manager functionality. It is of no use to users if the Tivoli Identity Manager Web application is available but they cannot perform required tasks on their accounts (such as a password reset) if the adapter is not available or not functioning as expected. There are various ways to monitor adapters. Consider the following aspects:

- ▶ Adapter availability: A particular adapter must be running in the first place for any account management operations to be performed on the relevant managed resource. Monitoring tools can be employed to connect directly to the adapter and check whether it is available at the network level. An extension of this is to use the Tivoli Identity Manager service definition *ping test* to check for the adapter's responsiveness via the Tivoli Identity Manager Web interface. Tracking the response time for the test may also potentially be useful, as this can indicate that performance is being affected. There may be various reasons for any degradation such as network latency or resource usage on the adapter's machine. These events can be manually investigated or may even be able to be determined by the monitoring system if it is tracking the types of events causing the issue.
- ▶ Adapter functionality: The availability of an adapter does not ensure functional availability. That is, account operations initiated by the Tivoli Identity Manager application and directed at the adapter may be received by the adapter, but it may not necessarily successfully process the request. This may be due to a multitude of factors. These events are tracked by both the Tivoli Identity Manager application audit trail and the adapter logs. Monitoring tools can be employed to watch for various Tivoli Identity Manager audit events, which may then result in the monitoring tool checking the adapter logs for a relevant error. Transaction IDs within Tivoli Identity Manager can be correlated between the Tivoli Identity Manager application to allow the monitoring tool to cross-reference the relevant adapter events. It may be the case that the monitoring tool is unable to diagnose the issue, but the functional issue is nevertheless reported to the monitoring application for action by a system administrator.

Identity feeds

These are usually scheduled to run at particular times without manual interaction. Identity feeds are also typically run with the assistance of IBM Tivoli Directory Integrator and require that the relevant Directory Integrator processes are running. This is especially critical when identity feeds are scheduled to run automatically without user intervention. Tivoli Directory Integrator comes with a set of monitoring features such as comprehensive logging and optionally sending event notifications via mail or as SNMP. It also provides a server notification system to monitor various processes taking place in the Tivoli Directory Integrator Server, such as AssemblyLine stop and start processes. Last but not least, Tivoli Directory Integrator bundles a Web-based Administration and Monitoring Console (AMC), as well as an Action Manager (AM). The combination of both applications can be used to monitor any Tivoli Directory Integrator processes and react on any (error) events fully automatically.

5.4.3 Sample monitoring tools and products

According to the complexity of monitoring concepts there is a large set available of either simple tools that address dedicated aspects only or comprehensive products that cover all Identity Manager components. It is out of scope of this discussion to provide an overview of all choices. However, we want to mention at least some of them.

Verbose Garbage Collection

If you are just interested in the WebSphere Application Server memory usage you could use the *Verbose Garbage Collection* to monitor JVM Heap Size. The Verbose Garbage Collection is a built-in feature of the WebSphere Application Server Java Runtime Environment and reports into local log files. Those log files can be parsed easily to show the relevant data only. Details on how to set up and use the tool are available at:

<http://www.ibm.com/developerworks/java/jdk/diagnosis/>

<http://www.ibm.com/support/docview.wss?uid=swg21114927>

Tivoli Performance Viewer

This Java application, shipped with WebSphere Application Server V6.1, takes monitoring feeds from the WebSphere Application Server Performance Management Interface (PMI) and presents the information in a graphical user interface. The configuration and use of PMI and the Tivoli Performance Monitor is described in at the following Web site:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tpvf_tpvmonitor.html

Once configured, the JVM heap usage is shown under the JVM Runtime viewer. Figure 5-11 shows a sample output on a Identity Manager system.

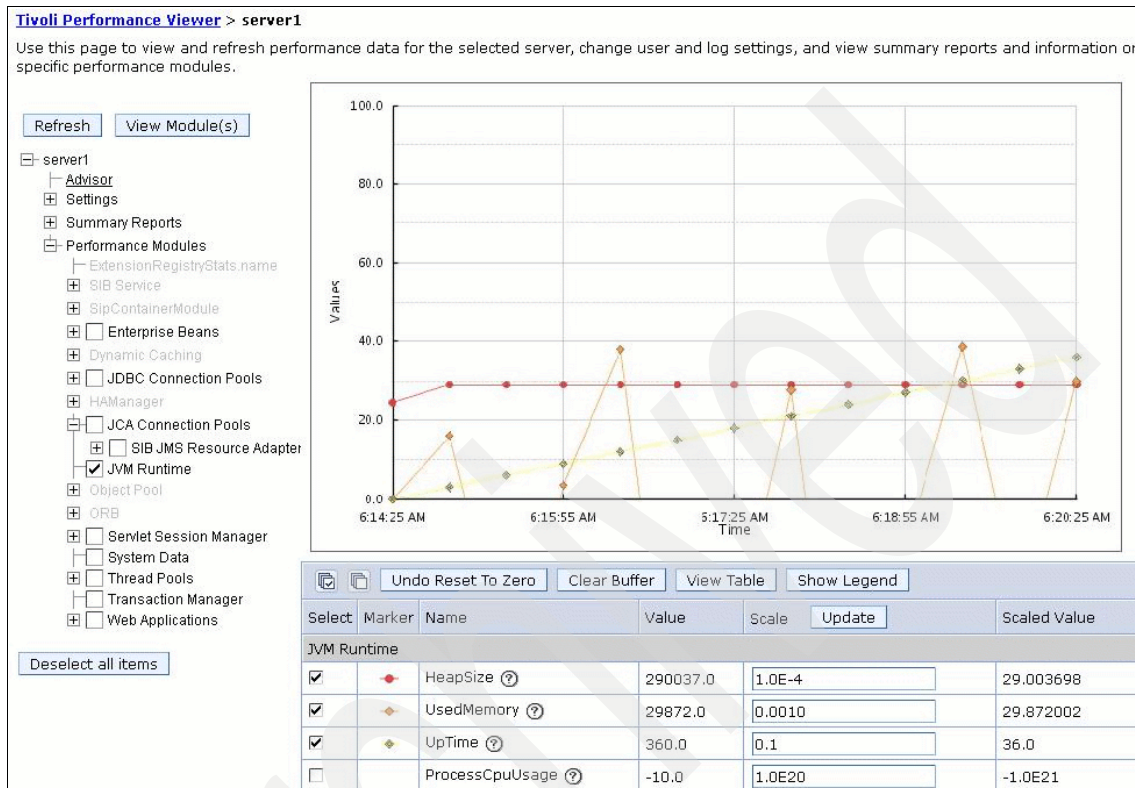


Figure 5-11 JVM Runtime viewer shows JVM heap usage

Furthermore, there are many third-party tools to monitor heap size and other performance metrics. There are also two tools available on alphaWorks® that take the garbage collection entries from the stderr log and produce a graphical representation:

- ▶ *Diagnostic Tool for Java Garbage Collector*
<http://www.alphaworks.ibm.com/tech/gcdiag>
- ▶ *IBM Pattern Modeling and Analysis Tool for Java Garbage Collector*
<http://www.alphaworks.ibm.com/tech/pmat>

Tivoli Monitoring and Universal Agent

A very comprehensive monitoring solution is represented by the IBM Tivoli Monitoring V6.1 product. There is a whitepaper available on OPAL describing a solution that has been developed to monitor Tivoli Identity Manager with the Tivoli Monitoring Universal Agent. Details can be found at:

<http://www.ibm.com/software/brandcatalog/portal/opal/details?catalog.label=1TW10TM40>

Some of the key questions that this solution addresses and that are important to any Tivoli Identity Manager deployment include:

- ▶ Is the Identity Manager server available?
- ▶ Is the server about to run out of memory?
- ▶ Are there any workflow processes backlogged?
- ▶ How responsive is the user interface?
- ▶ Is the audit database about to run out of space?
- ▶ Are there any errors being logged?

Figure 5-12 is taken from the OPAL whitepaper. It provides a sample view of Identity Manager database table spaces, JVM heapsize, CPU utilization, and workflow backlogs.

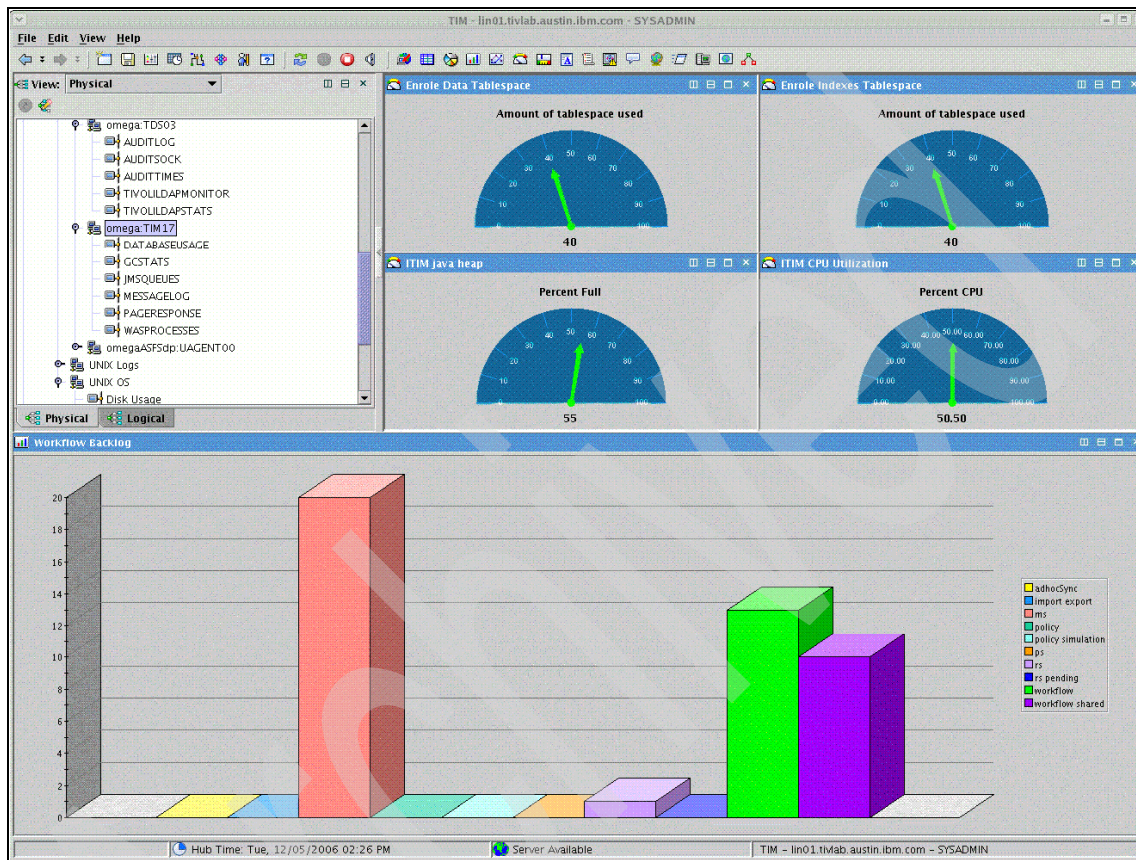


Figure 5-12 IBM Tivoli Monitoring for Tivoli Identity Manager

Tivoli Composite Application Monitoring

The IBM Tivoli Composite Application Monitoring solution is another example of a comprehensive monitoring solution.

The aim of IBM Tivoli Composite Application Manager is to simplify and enhance distributed application management as it is typical for Tivoli Identity Manager deployments. The Tivoli Composite Application Manager product family contains a cohesive set of tools to provide an end-to-end view of the application.

There are two members of the Tivoli Composite Application Manager suite that are of special interest in the context of this book. They both integrate with Tivoli Enterprise Portal mentioned above to visualize and correlate performance metrics and trends:

- ▶ *Tivoli Composite Application Manager for WebSphere* enables you to analyze the health of the WebSphere Application Server and the transactions that are invoked in it. It is able to trace the transaction execution to the detailed method-level information, connect transactions that spawn from one application server, and invoke services from other application servers.
- ▶ *Tivoli Composite Application Manager for Response Time* provides statistics of application transaction response times using instrumentation and robotic means. It enables you to analyze and break down end-to-end response time into individual components to quickly pinpoint a response time problem.

Some examples of the functions that the Tivoli Composite Application Monitoring solution provides are:

- ▶ Server statistics: JVM CPU and memory utilization, applications, and so on.
- ▶ Time-based activity tracking: Throughput, response time, sessions, users, and so on.
- ▶ Resource information: Thread pools, connection pools, locks, memory leak analysis, and so on.
- ▶ Server drill-downs: Allow for visual navigation of processes down to system component and Java method level to assist with identifying bottlenecks and determining problem root cause within an application.
- ▶ Performance analysis: Tracks application usage patterns and quality of service.
- ▶ Reporting: Produces analysis of the data collected.
- ▶ Alerts: Raised when monitored systems are not behaving as expected (for example, certain error or warning conditions are generated or transactions are falling outside of a predetermined acceptable response range).
- ▶ Playback components: Using its Synthetic Transaction Investigator, the solution allows for the ability to specify monitoring policies based on manual user activities by robotically executing user actions and recording these for playback by the monitoring application to automatically track a real business transaction for availability and performance. For example, a monitoring policy may be defined by recording the login process of a Web-based application through a Web browser, followed by subsequent navigation paths required to perform a specific business-driven application task and to have this recording used as the basis for availability and performance tracking/analysis.

For more information about the IBM Tivoli Composite Application Monitoring solution, refer to the IBM Redbooks publication *IBM Tivoli Composite Application Manager Family Installation, Configuration, and Basic Usage*, SG24-7151.

5.5 Security and integrity

The Tivoli Identity Manager software components allow flexibility to enable or disable various security features within each component and between the components. Many of these are generally disabled by default to allow for ease of development and initial deployment. As part of the design and planning, these configuration differences should be documented and decisions made with regards to which features must be enabled and in which environments. The production environment typically has the most security features enabled. While not a full exhaustive list, the main security considerations to note within an Tivoli Identity Manager environment are:

- ▶ Anonymous access to the LDAP: Typically disabled in production. That is, no anonymous access is allowed to the production LDAP. Other environments may or may not have this disabled depending on the perceived risks involved within the particular environment.
- ▶ Tivoli Identity Manager and LDAP communication via SSL: Typically configured for SSL in production. Other environments may or may not have this disabled depending on the perceived risks involved within the particular environment. If the LDAP is on the same machine as the Tivoli Identity Manager application, it may be decided that SSL is not required as the requests are not transmitted over the network.
- ▶ Tivoli Identity Manager and relational database communication via SSL: Not as common as SSL to the LDAP, but can be enabled between the database client and the database depending on the database being used. If the database is on the same machine as the Tivoli Identity Manager application, it maybe decided that SSL is not required, as the requests are not transmitted over the network.
- ▶ File permission settings on the Tivoli Identity Manager and application server specific file system directories: Dependent on various factors such as the accessibility of the machine, the users who can log on to the machine, the network segment that the machine is situated in, and the physical location of the machine.
- ▶ Passwords encrypted in Tivoli Identity Manager properties files: Typically enabled in production. This is an option that can be selected during installation. If disabled, passwords used to access Tivoli Identity Manager software components are stored in clear text within the configuration files.

- ▶ Pick a random password encryption key: The default value is given during installation. This is typically changed for production installations.
- ▶ Tivoli Identity Manager and adapter communication via SSL: Typically enabled in production. There is the option to configure the SSL communication to be one-way or mutually authenticated. Certain environments only require one-way SSL, while others with strict security policies may mandate that the SSL communications are mutually authenticated. If the adapter is on the same machine as the Tivoli Identity Manager application, it may be decided that SSL is not required, as the requests are not transmitted over the network.
- ▶ Tivoli Identity Manager and HR feed communication via SSL: Typically enabled in production. If the HR feed originates from the same machine as the Tivoli Identity Manager application, it may be decided that SSL is not required, as the requests are not transmitted over the network.
- ▶ Tivoli Identity Manager Web application access only via HTTPS: Enforces that users access the Tivoli Identity Manager application over a secure SSL channel. This requirement varies between deployments depending on the security policies of the organization.
- ▶ Do not allow non-local applications to connect to Tivoli Identity Manager: Relates to applications written to access Tivoli Identity Manager via the APIs. The Internet Inter-ORB Protocol (IIOP) communications between a J2EE client and server may be deemed to be insecure. Thus, an organization may mandate that all applications using the Tivoli Identity Manager API be local to the Tivoli Identity Manager application.
- ▶ Do not allow access to Tivoli Identity Manager's keystore: Relates to the private key used by Tivoli Identity Manager to encrypt information. This keystore is stored on the file system. Access to this keystore is typically limited to a select set of users in production. It may also be dependent on various factors such as the accessibility of the machine, the users who can log on to the machine, the network segment that the machine is situated in, and the physical location of the machine.
- ▶ File permission settings on each Tivoli Identity Manager adapter's specific file system directories: Dependent on various factors, such as the accessibility of the machine, the users who can log on to the machine, the network segment the machine that is situated in, and the physical location of the machine.
- ▶ Adapter login, password, and port configuration: Typically defaults to the same value on all adapters. The login, password, and port used by the Tivoli Identity Manager server to communicate with a specific adapter are stored within the adapter configuration settings. These are typically changed in a production environment.

- ▶ The agentCfg utility password: Most adapters use the agentCfg utility to make configuration changes to the adapter. The agentCfg utility uses a standard known default password, which is typically changed in a production environment.

Refer to the IBM Tivoli Identity Manager Information Center Version 5.1 at:

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

Also refer to adapter documentation for details on specifics on performing the above actions.

5.6 Conclusion

Operational and non-functional aspects are of high relevance during the design of identity management solutions. A thorough implementation of those requirements is crucial to comply with service level agreements (SLA), for example. Furthermore, high availability as well as reliability and performance have a direct impact on the consumer satisfaction, which builds the base of any successful identity management system.

In our experience after numerous Tivoli Identity Manager implementation projects of different scopes and complexity, we strongly recommend adhering to the following best practices:

- ▶ Operational and non-functional requirements must be considered from the very beginning of the solution design work. Doing the operational concept when getting into pilot phase testing is the wrong approach.
- ▶ Identity management systems always grow. At the time of designing a identity management it might not be possible to estimate how much and how quickly your system will grow, but it will. There are many reasons for that, such as growth of your company, providing service to new user groups like business partners and customers, integration of new target systems or additional workflow requirements. Therefore, design a solution that allows for horizontal as well as vertical scalability.
- ▶ Do not reinvent the wheel. The fact that Tivoli Identity Manager is based on industry standard middleware components like LDAP server, relational database, and WebSphere Application Server allows you to reuse existing concepts or integrate smoothly into given infrastructures. There is no need to implement operational concepts from scratch.

Archived

Tivoli Access Manager integration

In an identity and access management (IAM) solution, Tivoli Identity Manager and Tivoli Access Manager provide capabilities that enable you to better control *who* can access your protected IT systems and *what* kind of resources are available depending on users' business role memberships.

Access management addresses three questions from the business point of view:

- ▶ Who can come into my systems?
- ▶ What can they do?
- ▶ Can I easily prove what they have done with that access?

Using Tivoli Identity Manager and Tivoli Access Manager for e-business businesses can validate the authenticity of all users with access to resources and ensure that access controls are in place and consistently enforced.

This chapter describes how Tivoli Identity Manager both integrates with and manages aspects of Tivoli Access Manager. Topics covered include what management tools are available, what common components are used in an integrated environment, how to use Tivoli Access Manager to protect access to your Tivoli Identity Manager GUI interface, and how to synchronize accounts between both solutions and use all functionalities that Tivoli Access Manager provides.

6.1 Functional overview

Many customers need access and identity management together to help effectively manage parts of the security in their organization. Tivoli Identity Manager and Tivoli Access Manager are two products offered by Tivoli Software. This chapter describes how to use these products together to provide a complete management solution for users and their access to resources in an enterprise. Principal pieces of integration cover:

- ▶ Tivoli Identity Manager managing Tivoli Access Manager users
- ▶ Tivoli Access Manager for e-business protecting Tivoli Identity Manager user interface
- ▶ Data synchronization between Tivoli Access Manager user database and Tivoli Identity Manager

Tivoli Identity Manager can be used to manage Tivoli Access Manager users and groups. There are also different customizations that can be done to optimize the integration of Tivoli Identity Manager and Tivoli Access Manager when managing Tivoli Access Manager accounts. In order to manage Tivoli Access Manager accounts, a Tivoli Access Manager adapter must be installed and configured, and a Tivoli Identity Manager service is created to represent the Tivoli Access Manager user registry. A provisioning policy is then set up in Tivoli Identity Manager to enable provisioning Tivoli Access Manager accounts to users. Default values can be specified using account defaults or provisioning policy entitlement parameters to set Tivoli Access Manager attributes. Tivoli Access Manager account management can be automated with automatic provisioning, which creates Tivoli Access Manager accounts for every user with a given set of attributes. Reconciliation can be set up to import Tivoli Access Manager accounts into Tivoli Identity Manager. Tivoli Identity Manager views, access control items (ACIs), and groups can be created to implement a delegated administration hierarchy to allow administrative control over subsets of people.

IBM Tivoli Directory Integrator can be used for a Tivoli Identity Manager identity feed to import single-domain or multi-domain Tivoli Access Manager user data into Tivoli Identity Manager, import directory user data (for example, IBM Tivoli Directory Server) to Tivoli Identity Manager, and synchronize Tivoli Identity Manager user attributes with Tivoli Access Manager user attributes for automatic processing via reconciliations.

Each product is used for a different purpose, and is targeted towards different levels of administrators. Tivoli Access Manager is used to enforce access control, and an administrator would use the Tivoli Access Manager interfaces to set up a company's security policy to define categories (roles) of users who need access to different resources. Once the security policy is set up, then Tivoli Identity Manager can be used with its rich delegated administration capabilities to manage users and their access to resources.

Special considerations are also required to manage Tivoli Access Manager WebSEAL. Tivoli Identity Manager can manage business entitlements used by WebSEAL. WebSEAL can be used to protect Web applications. When Tivoli Identity Manager is deployed in a WebSEAL environment, the Tivoli Identity Manager Web application is protected as well. In this environment, a user authenticates to Tivoli Access Manager so that WebSEAL can determine whether that user is authorized to access specific Web applications.

Single sign-on can be set up between Tivoli Identity Manager and Tivoli Access Manager so that users do not have to authenticate to both products. There are some implications when setting up single sign-on between those components. Tivoli Identity Manager provides a self-care solution so that users can reset passwords via a Web browser. With Tivoli Access Manager protecting access to Tivoli Identity Manager, you must consider which pages will be granted anonymous access in order to allow users to reset their passwords via challenge/response. Another way to allow users to change their passwords is the use of the *pkmspasswd* function of Tivoli Access Manager, but this approach does not provide the challenge/response capability to users. For more details, see the *IBM Tivoli Access Manager for e-business WebSEAL Administration Guide Version 6.1*, SC23-6505.

Since Tivoli Identity Manager performs central user administration tasks, it can perform user password synchronization across platforms or applications. The password synchronization function usually operates from Tivoli Identity Manager to all adapters configured. This means that password changes should be primarily performed using the Tivoli Identity Manager Web user interface. If you want to allow users to change their passwords using the Tivoli Access Manager *pkmspasswd* function, be aware that the standard reconciliation process does not pick up these password changes. You must install the *Tivoli Access Manager Password Synchronization Adapter*, which is available as part of the Tivoli Access Manager Combo Adapter package¹.

¹ The Tivoli Access Manager Combo Adapter package is available as part of the Tivoli Identity Manager offering. The *Tivoli Identity Manager - Tivoli Access Manager Combo Adapter Installation and Configuration Guide Version 5.1*, SC23-9664, is available in the Tivoli Identity Manager V5.1 Information Center in the Adapter Documentation section at:
<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

6.2 Managing Tivoli Access Manager with Tivoli Identity Manager

Tivoli Identity Manager can manage a user's data, along with a user's provisioning policy, which contains entitlements that represent a user's accounts and access on various systems and applications. Setting up automatic provisioning can create an access manager account automatically when a user becomes a member of an organizational role. It can also manage manual provisioning of Tivoli Access Manager accounts and group accesses using the request account and request access functionality.

Once Tivoli Access Manager entitlements are set up as part of a user's provisioning policy and the appropriate accesses have been defined, Tivoli Identity Manager administrators can manage Tivoli Access Manager accounts and accesses for that user. Tivoli Identity Manager also allows Tivoli Access Manager customization in the form of entitlement attributes, identity policy, recertification policy, and password policy.

One additional feature that should be considered when managing Tivoli Access Manager data is Tivoli Identity Manager's delegated administration capability.

6.2.1 Creating a Tivoli Access Manager service

In order to start managing accounts in a Tivoli Access Manager deployment, the Tivoli Access Manager combo adapter and service profile must be deployed. Then the Tivoli Access Manager service must be defined.

The Tivoli Access Manager combo adapter leverages the IBM Tivoli Directory Integrator server to facilitate communication between Tivoli Identity Manager and Tivoli Access Manager.

Using the combo adapter, you can automate the following administrative tasks:

- ▶ Creating new users for Tivoli Access Manager
- ▶ Creating single sign-on (SSO) credentials for users on Tivoli Access Manager
- ▶ Modifying users' SSO credentials and attributes on Tivoli Access Manager and its underlying Lightweight Directory Access Protocol (LDAP) server
- ▶ Changing user account passwords on Tivoli Access Manager
- ▶ Suspending, restoring, and deleting user accounts on Tivoli Access Manager
- ▶ Reconciling user, SSO credentials, and LDAP user attributes on Tivoli Access Manager

Installing the Tivoli Access Manager adapter and adapter profile

Information about how to install the Tivoli Access Manager adapter and import the adapter profile into the Tivoli Identity Manager server can be found in the book *Tivoli Identity Manager - Tivoli Access Manager Combo Adapter Installation and Configuration Guide Version 5.1*, SC23-9664. After the adapter has been installed on the machine where Tivoli Directory Integrator is located, the Tivoli Access Manager JAVA run time environment (AMJRTE) must also be installed and configured. Once this is done the adapter can be registered into the Tivoli Access Manager secure domain using the SvrSslCfg utility. As with other Tivoli Identity Manager adapters, after the adapter has been installed and configured, the SSL certificates must be installed. The installation and configuration guide describes how to configure the above steps in detail.

In a Tivoli Access Manager for Operating Systems environment, multiple adapters may be required on the same machine—one for Tivoli Access Manager and one to manage the UNIX user registry. Different port numbers are needed for each of the adapters.

Creating the Tivoli Access Manager service

Before Tivoli Identity Manager can manage accounts in Tivoli Access Manager, a Tivoli Access Manager service must be created. To create a service, the administrator must navigate to the Manage Services task in the Tivoli Identity Manager GUI. Clicking the **Create** button shows a list of available service types that can be added. Select the **Tivoli Access Manager Combo Profile** service type. If the Tivoli Access Manager Combo Profile service type is not listed, then the adapter profile installation was either not performed, the application server has not refreshed its cache, or the import was not successful. Enter the required attributes in the Service Setup, Tivoli Access Manager Setup, and LDAP Setup tabs of the Tivoli Access Manager service to complete the task.

6.2.2 Setting up a Tivoli Access Manager specific policy

Another aspect of managing Tivoli Access Manager account types is setting up a policy specific to Tivoli Access Manager. The different types of policy to consider are password policy, identity policy, and provisioning policy. Password policy specifies the password rules that control the content of an account password when a user changes his password. Identity policy defines how to create user IDs for accounts. The provisioning policy is responsible for the automated provisioning and de-provisioning of user accounts into the Tivoli Access Manager environment.

Identity policy

An identity policy defines how a user's ID is created. Tivoli Identity Manager automatically generates user IDs from the identity policy, which can be assigned per service or service type. Tivoli Identity Manager allows JavaScript code to be written to specify the string to use for the user ID.

For Tivoli Access Manager, one method of creating a user ID is to base it on a user's common name (CN) and her surname or last name (SN). Example 6-1 shows a very simple JavaScript that could be used as part of a Tivoli Access Manager service identity policy.

Example 6-1 Tivoli Access Manager service identity policy JavaScript

```
function createIdentity()
{
    var cn = subject.getProperty('cn')[0];
    var sn = subject.getProperty('sn')[0];
    var userid = "";

    userid = cn.toLowerCase().substring(0,1) + sn.toLowerCase();

    return userid;
}
return createIdentity();
```

A JavaScript object representing the person entity within the Tivoli Identity Manager data model is referred to as the *subject*. In this example, the *cn* and *sn* are used from the person entity to generate the user ID. The first character of the *cn* is appended to the last name to create the user ID. The user ID is returned in lower case.

Be aware that Example 6-1 is very simplistic. It does not handle any special circumstances, for example, the case of overlapping user IDs. In any real-world deployment you must include enhancement before this identity policy could be used. For a more complex identity policy example refer to the IBM Redbooks publication *Deployment Guide Series: IBM Tivoli Identity Manager 5.0*, SG24-6477.

Tip: The script above was autogenerated by the *Simple - define rule* selection and then slightly modified. For any identity policy changes that cannot be accomplished by the simple definitions, it is still best to create a simple definition that is as close as possible to the desired rule (this creates the body of the script), then use the advanced view to edit in the finishing touches.

Password policy

Considerations should also be made in regards to the password policy. There are three areas of password policy that should be addressed:

- ▶ Password synchronization
- ▶ Password strength policy
- ▶ Password login policy

Tivoli Identity Manager provides the capability to keep passwords synchronized. There is a property that can be set from the Tivoli Identity Manager GUI that enables password synchronization, as shown in Figure 6-1.

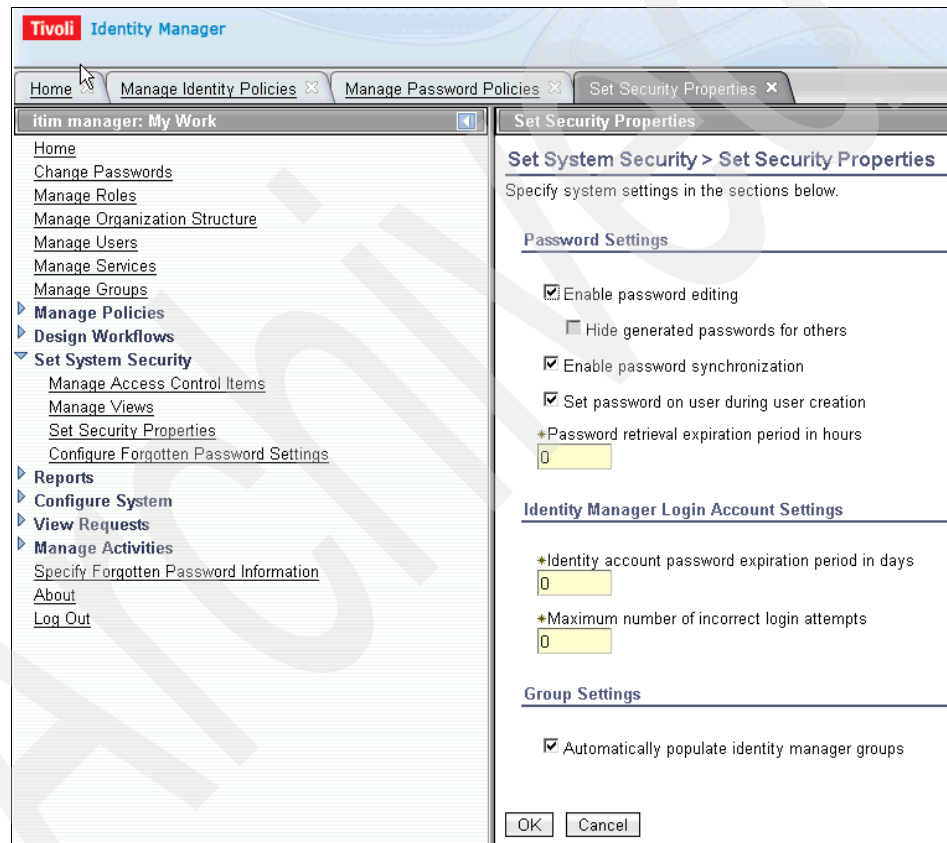


Figure 6-1 Password synchronization

To enable this feature, go to the **Set System Security → Set Security Properties portfolio** task and check **Enable password synchronization**. If this feature is enabled, Tivoli Identity Manager keeps all of a user's passwords the same. Users will not be able to change account passwords separate from their Tivoli Identity Manager password using the Tivoli Identity Manager interface.

Tivoli Identity Manager also provides the capability to set a *password strength policy*, which Tivoli Identity Manager calls the password policy, at a variety of levels. Password policy can be set at a service level, a service type level, or for all services.

In order to keep passwords synchronized between Tivoli Identity Manager and Tivoli Access Manager, all password change operations should go through Tivoli Identity Manager, and Tivoli Identity Manager can both enforce a common password policy and keep the passwords synchronized. Users should not be allowed to change their passwords through the `pdadmin` command or the Web Portal Manager Web interface. Note that the password policy must be the same on Tivoli Identity Manager and Tivoli Access Manager, because a password set on Tivoli Identity Manager could fail when it is being set up on the Tivoli Access Manager system because it does not match the password rules on the Tivoli Access Manager system.

If WebSEAL has been deployed in the environment, it also has the capability of changing a user's Tivoli Access Manager password. In this environment, the WebSEAL reverse password synchronization module should be installed into WebSEAL so that all password checking goes through Tivoli Identity Manager, and the Tivoli Access Manager password is synchronized with Tivoli Identity Manager managed passwords. Refer to "Reverse password synchronization" on page 287 for more information.

In a Tivoli Access Manager for Operating Systems environment, the best method of keeping passwords synchronized is to have users change their passwords through the Tivoli Identity Manager Web Interface. This allows Tivoli Identity Manager to manage the passwords in Tivoli Access Manager as well as on the different UNIX endpoints. Measures can be taken to disable the change password operation on the different UNIX machines. However, when a password has expired, some users have no way to get to the Web interface because they cannot log into their systems. The Tivoli Access Manager for Operating Systems login policy can be set to include a number of grace logins, which would allow a user to log onto a system after the expiration date. Then the user would have the opportunity to perform a change password operation using a browser without being locked out of the system.

Provisioning policy with Tivoli Access Manager entitlements

Before Tivoli Access Manager accounts can be created for users, an appropriate provisioning policy must be set up with Tivoli Access Manager entitlements. Provisioning policies apply to organizational roles and can be configured for automatic or manual entitlements, with some specific details. These details depend on the type of service, and there is a parameter list for each type.

The Tivoli Access Manager profile provides specific attributes to configure. You can define entitlement parameters in provisioning policies or account defaults on the service type or at a specific service to automatically fill in those attributes. The attributes can either be set as constant values, calculated using JavaScript, or set as default values. Refer to IBM Tivoli Identity Manager Information Center Version 5.1 for more information and examples at:

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

Figure 6-2 is an example of entitlement parameters set in a provisioning policy with specific attributes configured via JavaScript or as a constant.

Select	Name	Template value	Enforcement T	Value Type
<input type="checkbox"/>	Single Signon Capability	true	Default	Constant Value
<input type="checkbox"/>	Distinguished Name	"cn="+subject.getProperty("cn")[0]+"o,=tam,c=us"	Default	JavaScript

Page 1 of 1 Total: 2 Displayed: 2 Selected: 0

Figure 6-2 Entitlements parameter example for Tivoli Austin Airlines

Tivoli Identity Manager provides two options to set account defaults. One is based on the account type, defined globally for all services of the specific profile. The other one is specific to the service definition.

Figure 6-3 is an example of default values being set using account defaults, based on the service type.

Configure System > Manage Account Defaults > Select an Account Attribute

You can add, change, or remove attribute defaults. To change a default for the **TAM Combo Profile** service type, select the attribute in the table, and then click the appropriate button.

Select	Account attribute	Template value
<input type="checkbox"/>	Full name	{Full name}
<input type="checkbox"/>	Last name	{Last name}

Page 1 of 1 Total: 2 Displayed: 2 Selected: 0

Figure 6-3 Account defaults for Tivoli Access Manager accounts

Figure 6-4 is an example of default values being set using account defaults, based on the service definition. The service name for this definition is “Access Manager for .NET Banking App”.

Manage Services > Manage Account Defaults > Select an Account Attribute

You can add, change, or remove attribute defaults. To change a default for the **Access Manager for .NET Banking App** service, select the attribute in the table, and then click the appropriate button.

Use the global account defaults for the service type

Select	Account attribute	Template value
<input type="checkbox"/>	Distinguished Name	Script provided
<input type="checkbox"/>	Full name	{Preferred user ID}
<input type="checkbox"/>	Last name	{Last name}

Page 1 of 1 Total: 3 Displayed: 3 Selected: 0

Figure 6-4 Account defaults for Tivoli Access Manager accounts on a specific service

Creating a Tivoli Access Manager account

To create a Tivoli Access Manager account for a user:

1. Go to the Manage Users task in the navigation tree.
2. Locate the user by typing information about the user in the Search Information field, select a filter, and then click the **Search** button.
3. From the users drop down menu, select the **Request Accounts** user task.
4. Locate the Tivoli Access Manager service by typing information about the service in the Search Information field, and then click the **Search** button.

5. If the Tivoli Access Manager service does not appear in the list of available services, then either the provisioning policy was not set up correctly or the membership was not specified to include the user who is being managed. Select the **Access Manager** service and click the **Continue** button.
6. The Request an Account form should have the user ID filled in via the identity policy, and the full name and the last name filled in via the defined account defaults. JavaScript in the entitlement parameters of the Tivoli Access Manager provisioning policy will fill in the LDAP dn, as shown in Figure 6-5.

Manage Users > Request an Account > TAM Account	
Type the appropriate information for the account. When you are done specifying information on each of the tabs, click Continue.	
*User ID	<input type="text" value="jsmith"/>
*Distinguished Name	<input type="text" value="cn=James Smith,o=tam,c=us"/>
*Full name	<input type="text" value="James Smith"/>
*Last name	<input type="text" value="Smith"/>
Description	<input type="text"/>
<input type="button" value="Continue"/> <input type="button" value="Cancel"/>	

Figure 6-5 Tivoli Access Manager user information

7. In the Tivoli Access Manager Policy tab, group membership can be defined. A full reconciliation or reconciliation of supporting data must be performed to get all the group names into Tivoli Identity Manager. Group membership is discussed below.
8. In the SSO Resource tab, SSO credentials can be defined for Tivoli Access Manager GSO resources.
9. Press **Continue**, specify the password, and then press **Submit**.

6.2.3 Managing group memberships for Tivoli Access Manager

Administrators can also manage Tivoli Access Manager groups' memberships using Tivoli Identity Manager. Tivoli Access Manager groups can be managed by modifying the LDAP group membership attribute on a user's account. To add a user to a group, select the Tivoli Access Manager account from a user's account list, and add the group to the user's group membership attribute in the Tivoli Access Manager Policy tab. To delete the user from a group, remove that group from his group membership attribute. Tivoli Identity Manager obtains the list of Tivoli Access Manager groups when reconciliation is performed against a Tivoli Access Manager service. Regularly scheduled reconciliation must be performed to keep the group list up to date. Tivoli Access Manager groups cannot be created or deleted using the Tivoli Identity Manager interface. For that purpose either Web Portal Manager or `pdadmin` must be used.

6.2.4 Reconciliation

As with other Tivoli Identity Manager services, Tivoli Access Manager accounts managed by Tivoli Identity Manager can be kept in sync with Tivoli Access Manager through reconciliation.

When setting up reconciliation for Tivoli Access Manager, certain system accounts must be excluded, and should not be adopted into the Tivoli Identity Manager database. Accounts can be excluded from automatic adoption based on the profile of the service. To exclude accounts, the account names must be entered into the LDAP directory that is used for the Tivoli Identity Manager database. The accounts can be entered manually, or an LDAP Data Interchange Format (LDIF) file can be used. Example 6-2 show a sample LDIF file that can be used to specify the accounts to exclude from Tivoli Access Manager.

Example 6-2 LDIF example

```
dn: ou=excludeAccounts,ou=itim,ou=taa,dc=com
ou: excludeAccounts
objectClass: top
objectClass: organizationalunit

dn: cn=itamprofile,ou=excludeAccounts,ou=itim,ou=taa,dc=com
erObjectProfileName: itamprofile
objectClass: top
objectClass: eridentityexclusion
cn: itamprofile
erAccountID: sec_master
erAccountID: ivmgrd/master
erAccountID: ivacld/ServerName
erAccountID: amwpm/ServerName
erAccountID: default-webseald/ServerName
```

This LDIF file creates the `excludeAccounts` container object in LDAP, and then adds an entry for *itamprofile*, which applies to all services of type *itamprofile*. The excluded list contains several accounts. Tivoli Access Manager creates the `sec_master` account for administrative purposes, and that account should not be reconciled into Tivoli Identity Manager. The other accounts are used as server principals, and are not assigned to people. The `cn` and `erObjectProfileName` represent the name of the service profile. Excluded accounts are defined by the `erAccountID` attribute. The example excludes `sec_master`, `ivmgrd/master`, `ivaclD/ServerName`, `amwpm/ServerName`, and `default-webseald/ServerName` accounts from automatically being adopted when a reconciliation is performed on a Tivoli Access Manager service.

To get an exact list of accounts and their exact names, the `pdadmin` command should be used to list all the users:

```
pdadmin> user list * 100
sec_master
ivmgrd/master
ivaclD/ServerName
amwpm/ServerName
default-webseald/ServerName
```

Reverse password synchronization

As mentioned in 6.1, “Functional overview” on page 276, the Tivoli Access Manager Adapter for Tivoli Identity Manager provides integration between Tivoli Access Manager WebSEAL and Tivoli Identity Manager. The adapter provides synchronization in both directions between Tivoli Identity Manager and Tivoli Access Manager for all user attributes except user passwords. These are only synchronized in one direction, from Tivoli Identity Manager to Tivoli Access Manager. To achieve synchronization of user passwords from Tivoli Access Manager to Tivoli Identity Manager, the *password synchronization adapter* must be installed.

The password synchronization adapter has two basic components:

- ▶ A server-side component, installed on the Tivoli Identity Manager Server
- ▶ A client-side component, installed on the Tivoli WebSEAL Server

The server-side component is installed with the Tivoli Identity Manager server. Both of these components must be installed before the Tivoli Identity Manager Server will accept password changes from the WebSEAL password change Web page, *pkmspasswd*. The adapter only synchronizes passwords changed through this page and after it has checked the password strength as defined in the Tivoli Identity Manager password policy.

When a user successfully changes his password using the WebSEAL password synchronization adapter, Tivoli Identity Manager changes all passwords for managed targets with accounts for that user if the password synchronization feature is enabled in Tivoli Identity Manager.

6.2.5 Automatic provisioning

When a new user joins the organization, Tivoli Identity Manager can provision accounts for this user either manually or automatically. For automatic provisioning *organizational roles* can be set up so that users are either dynamically given membership depending on values of user attributes (dynamic roles) or manually given membership by an administrative person (static roles). These roles can be attached to a *provisioning policy* with Access Manager entitlements, and this policy can provision accounts to Tivoli Access Manager automatically.

An entitlement can also specify only group membership, which can be used to automatically add a user to a Tivoli Access Manager group when that user becomes a member of a given organizational role.

With the appropriate organizational roles defined and provisioning policies set to manual, the *request account* or *request access* functionality can also be used so that users can manually request Tivoli Access Manager accounts and access entitlements.

6.2.6 Delegated user administration

The Tivoli Access Manager Web Portal Manager (WPM) provides capabilities to support delegated administration. Customers can use WPM to create delegate domains and give administrators of these domains control over the user management of those domains. When a domain is created, several administrator roles are defined so that different types of administrators can manage users in the domain. Delegate administration provides a Tivoli Access Manager administrator with the capability to create delegate user domains, create new users, add existing users to additional domains, and assign various types of administrators to the domains. These delegate administrators can then perform a subset of administration functions, depending on their type, on the users in their assigned domain.

A delegated administration domain has different types of administrators that can perform different levels of administration. There are five types of administrator roles defined in Tivoli Access Manager:

- ▶ Tivoli Access Manager administrator
- ▶ Domain administrators
- ▶ Senior administrators
- ▶ Administrators
- ▶ Support administrators

The Tivoli Access Manager administrator is a member of the *iv-admin* group. This administrator can perform all delegate administration functions. For user management and delegate purposes the other types of administrators are used.

Tivoli Identity Manager provides similar but more powerful functionality. By default, the following provided components can be used for delegation:

- ▶ Admin domains

Administrative domains are logical collections of resources that are used to separate responsibilities and manage permissions. An admin domain identifies a subsidiary part of an organization as a separate entity with its own policies, services, and access control items, including an administrator (also called domain administrator) whose actions and views are restricted to that domain.

- ▶ Tivoli Identity Manager groups

Tivoli Identity Manager provides predefined groups that have associated views and access control items and that can be used for delegation of administrative tasks. The predefined groups are:

Administrator	The administrator group has no limits set by default views or access control items and can access all views and perform all operations in Tivoli Identity Manager. The first system administrator user is named <i>itim manager</i> .
Service owner	Members of the service owner group manage a service, including the user accounts and requests for that service.
Manager	Members of the manager group are users who manage the accounts, profiles, and passwords of their direct subordinates.
Auditor	Members of the auditor group can request reports for audit purposes.
Helpdesk	Members of the helpdesk assistant group can request, change, suspend, restore, and delete accounts.

Members can request, change, and delete access, and also can reset others' passwords, profiles, and accounts. Additionally, members can delegate activities for a user.

Using Tivoli Identity Manager admin domains along with Tivoli Identity Manager groups, views, and access control items, delegation hierarchies can be set up with similar capabilities to the Tivoli Access Manager WPM delegated administration domains.

6.3 Resource versus user management

When managing a security infrastructure, there are administrators that take on different roles that define their administrative duties. Tivoli Access Manager and Tivoli Identity Manager provide tools to manage secure access to enterprise resources. However, the different types of administrators manage different aspects of the security infrastructure and take on different roles. Tivoli Access Manager tools can be used to manage the secure access to resources, and Tivoli Identity Manager can be used to manage users and their access to resources.

6.3.1 Different types of administrators

Administrators that manage the security of an enterprise perform different tasks in various roles when they implement the security policy of a company. One security administration role is that of setting up a company's security policy. Administrators in that role determine what resources must be protected and what groups of users would need access to those resources. This type of administrator, the security policy administrator, does not know which users get access to resources, so he would determine which types of users get access to resources, and would set up resource access lists or security roles to protect those resources. A second security administration role is identity management. An administrator performing tasks in that role deal more with day-to-day activities, create user identities, and give users access to resources. This category of administrator may not understand how to set up the resource security policy, but knows what resources are available and how to give users access to those resources.

When managing users and resources using Tivoli Identity Manager and Tivoli Access Manager, different interfaces are available for the two administration roles described. A security policy administrator uses the Tivoli Access Manager Web Portal Manager to manage protected resources and to create access control lists with appropriate groups on the ACLs. The groups represent the different types of

users that need access to resources. Once this policy is set up that administrator also creates the appropriate Tivoli Identity Manager organizational roles and provisioning policy to represent the different groups, setting up the Tivoli Identity Manager environment for the user management administrator. The user management administrator uses the Tivoli Identity Manager interfaces to create user identities and Tivoli Access Manager accounts and to put users in Tivoli Identity Manager organizational roles that are attached to provisioning policy with entitlements to Tivoli Access Manager groups, thus giving them access to Tivoli Access Manager protected resources.

6.3.2 Resource management

Tivoli Access Manager Web Portal Manager is a Web-based application that is used to manage Tivoli Access Manager resources. Tivoli Access Manager also has the pdadmin administration utility, which provides a full set of Tivoli Access Manager management tasks, but only through a command-line interface. WPM provides tasks to manage all Tivoli Access Manager objects used to implement security policy, such as groups, the protected object space, ACLs, action groups, protected object policies (POPs), and global sign-on (GSO) resources.

WPM can also handle the tasks to manage Tivoli Access Manager users and groups. However, these tasks, along with the WPM delegated administration support, should be addressed using the equivalent Tivoli Identity Manager functionality. Tivoli Identity Manager provides a *unified capability* to manage Tivoli Access Manager users along with other types of user accounts. If WPM is used to manage the user aspect of Tivoli Access Manager, then the full power of Tivoli Identity Manager cannot be realized. WPM also allows users to be assigned to Tivoli Access Manager groups. Again, Tivoli Identity Manager should be used to manage Tivoli Access Manager groups so that Tivoli Identity Manager can keep track of which users are in which groups.

6.4 Integration with Tivoli Access Manager WebSEAL

Two different aspects of integration must be considered when using Tivoli Access Manager WebSEAL. One feature that Tivoli Identity Manager can help manage is the WebSEAL capability to insert LDAP tag value pairs into a Web interaction. Another scenario that must be considered is when WebSEAL is protecting the Web server that Tivoli Identity Manager is using as a GUI server.

6.4.1 Managing LDAP attributes used by WebSEAL

WebSEAL has a feature that enables *dynamic business entitlements*, also known as *tag/value support*. This feature allows entitlement data to be included in a user credential so that the data can be used later for an access control decision. This entitlement data can be pulled out of LDAP where it is stored in the form of attributes on the person object associated with the Tivoli Access Manager user. The person object can be extended to add additional attributes to implement company-specific entitlements.

Tivoli Identity Manager and Tivoli Access Manager maintain their own separate person objects in the directory. The Tivoli Access Manager Combo Adapter provided with Tivoli Identity Manager supports the management of not only standard Tivoli Access Manager attributes and GSO credentials, but also attributes in the *inetOrgPerson* object class. If you are not using the *inetOrgPerson* object class, and your objectclass has an attribute that is not an *inetOrgPerson* standard attribute, you can also customize the combo adapter to accommodate your requirements. Chapter 3, “Installing the Tivoli Access Manager Combo Adapter”, in the *Tivoli Identity Manager - Tivoli Access Manager Combo Adapter Installation and Configuration Guide Version 5.1*, SC23-9664, describes this in more detail.

Since Tivoli Identity Manager allows the management of person attributes on the Tivoli Access Manager user object, a combination of Tivoli Identity Manager, Tivoli Access Manager, and the combo adapter can therefore be used to manage company-specific entitlements through the Tivoli Identity Manager interface, as depicted in Figure 6-6.

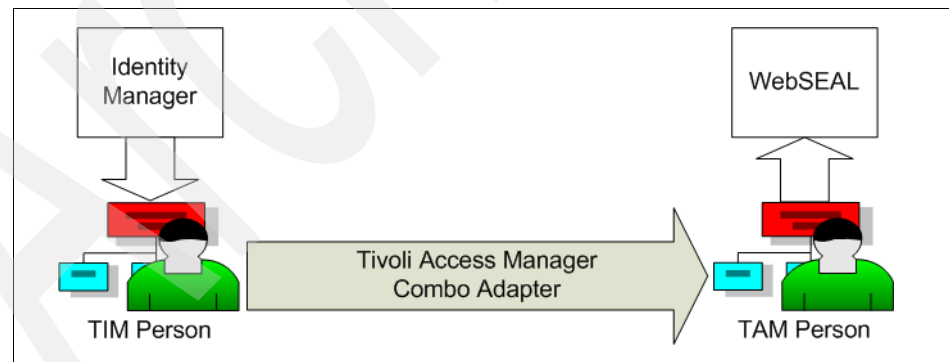


Figure 6-6 Synchronizing Tivoli Identity Manager attributes with Tivoli Access Manager entitlements

Tivoli Identity Manager is designed to be the focal point for corporate identity management. However, in your environment, other IBM Tivoli security applications with user management (such as Tivoli Access Manager) might have already been installed and might coexist with Tivoli Identity Manager. Therefore, several user data records might exist for the same user. Because Tivoli Identity Manager requires its own user registry and it cannot share the user objects that are in the user registry of another application (such as Tivoli Access Manager or a corporate directory), you must create new user records in Tivoli Identity Manager or import existing user data records from other data resources to Tivoli Identity Manager if you want Tivoli Identity Manager to manage those users. If Tivoli Access Manager or other applications with user data records coexist with Tivoli Identity Manager and up-to-date user attributes are needed for these applications, Tivoli Identity Manager data must be dynamically synchronized with the user records in these applications.

As a part of the installation process, a directory is created at the Tivoli Identity Manager Server system containing some extensions to use with the solution. There is a set of files in the folder <ITIM install directory>/extensions/5.1/examples/idi_integration/tam, containing four AssemblyLines samples. These AssemblyLines are used in IBM Tivoli Directory Integrator to import Tivoli Access Manager and corporate directory users to Tivoli Identity Manager and to synchronize Tivoli Identity Manager user attributes with those in Tivoli Access Manager. (Note that this example does not utilize the combo adapter mentioned earlier).

Directory Integrator can use different connectors to retrieve Tivoli Access Manager user data or corporate human resources data from a registry server and then feed it to Tivoli Identity Manager.

The main functions of the four provided AssemblyLines include:

- ▶ Importing Tivoli Access Manager users (in a single domain) into Tivoli Identity Manager
- ▶ Importing Tivoli Access Manager users (in a multi-domain) into Tivoli Identity Manager
- ▶ Importing users from an existing corporate directory into Tivoli Identity Manager
- ▶ Synchronizing Tivoli Identity Manager user attributes with Tivoli Access Manager user attributes

Tivoli Directory Integrator is designed to synchronize identity data located in directories, databases, collaborative systems, applications used for human resources, customer relationship management (CRM), Enterprise Resource Planning (ERP), and other corporate applications.

For more information, see the file <ITIM install directory>/extensions/5.1/examples/idi_integration/tam/tamimportsync.html on the Tivoli Identity Manager server. Also see the section “Identity feed management” in the IBM Tivoli Identity Manager Information Center Version 5.1, available at:

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

6.4.2 Setting up Web single sign-on

A common architecture used by many customers is to have a protected area within their network called the demilitarized zone (DMZ), where firewalls are used to control access from the external Internet to the internal intranet. The Tivoli Access Manager for e-Business component for this purpose is WebSEAL. The only access inside the DMZ is to the WebSEAL reverse proxy server, which controls any access to the internal Internet. Tivoli Identity Manager may be set up inside the internal intranet, with WebSEAL controlling access to the Tivoli Identity Manager GUI. When users access WebSEAL for the first time, a Tivoli Access Manager authentication is required (for example, the Tivoli Access Manager account user ID and password are entered).

With this type of setup, it would be nice to have single sign-on (SSO) enabled so that an administrator only enters a user ID and password once.

In setting up an SSO environment between Tivoli Access Manager WebSEAL and Tivoli Identity Manager, several steps are required to customize Tivoli Identity Manager and to set up WebSEAL, which we discuss in the following section.

Configuring Tivoli Identity Manager for SSO

Single sign-on can be configured for both Tivoli Identity Manager administrative console and self-service console applications using Tivoli Access Manager. With the single sign-on function configured a user logs in once and authenticates to a Tivoli Access Manager Web security server. From there the user's identity is propagated to the Tivoli Identity Manager application, eliminating the need for another login.

This function requires Tivoli Access Manager to be enabled for the single sign-on capability with Tivoli Identity Manager:

1. Tivoli Access Manager performs user authentication and coarse-grained authorization before access is allowed to Tivoli Identity Manager.
2. Tivoli Identity Manager then applies fine-grained access control using its own access control item.

There are two ways to configure a Tivoli Access Manager Web security server and Tivoli Identity Manager for single sign-on:

- ▶ Using WebSEAL
- ▶ Using a Tivoli Access Manager plug-in server

The overall picture for setting up single sign-on includes the following steps:

1. Configure the HTTP sever. The actual procedures vary depending on whether Tivoli Access Manager WebSEAL or a Tivoli Access Manager Plug-in is being used to provide the single sign-on capabilities.
2. Configure the Tivoli Access Manager Web security server to insert the user's identity in traffic that is passed to Tivoli Identity Manager.
3. Modify the Tivoli Identity Manager properties file to enable single sign-on.
4. Apply Tivoli Access Manager authorization to the Tivoli Access Manager protected object space so that only authorized users are allowed access to Tivoli Identity Manager.

For more information see the section “Configuring single sign-on” in the IBM Tivoli Identity Manager Information Center Version 5.1, available at:

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

In order to configure Tivoli Identity Manager for SSO using a custom login interface, look at the single sign-on examples in the document <ITIM install directory>/extensions/5.1/doc/singlesignon/Readme.html available on your Tivoli Identity Manager server.

6.5 Tivoli Access Manager for Operating Systems

Since Tivoli Access Manager for Operating Systems uses base Tivoli Access Manager functionality, Tivoli Identity Manager can manage Tivoli Access Manager for Operating Systems users as well. Tivoli Access Manager for Operating Systems users are the same as Tivoli Access Manager users. However, Tivoli Access Manager for Operating Systems has an additional requirement that Tivoli Identity Manager can help to manage. Tivoli Access Manager for Operating Systems requires that the Tivoli Access Manager user ID or short name be synchronized with the user's UNIX user ID or short name where Tivoli Access Manager for Operating Systems is running.

Managing Tivoli Access Manager accounts in sequence with UNIX

In order to manage the Tivoli Access Manager accounts with the corresponding UNIX accounts, we recommend the following steps:

1. Create an identity policy that applies to the Tivoli Access Manager service and to the UNIX services.
2. Create a provisioning policy with entitlements that creates accounts in Tivoli Access Manager and on the UNIX machines at the same time.

See “Identity policy” on page 280, which tells you how to create the JavaScript needed for identity policies. When creating the identity policy for UNIX and Tivoli Access Manager accounts, make sure to select the appropriate service types on the Services tab in the Define Identity Policy task. Actual service instances can be specified if the identity policy should only apply to specific UNIX machines.

When creating a provisioning policy for UNIX and Tivoli Access Manager accounts, entitlements must be specified to create the appropriate accounts. There are several options that can be used to create the proper entitlements. One entitlement can be created for the Tivoli Access Manager account, and one for each UNIX machine. Another option would be to create one for the Tivoli Access Manager account, and one for all machines of a specific target type. There are target type profiles defined for each type of UNIX (for example, AIXProfile, SolarisProfile, and HpuxProfile). The entitlements should be set up to provision automatically so that when a user is placed, either automatically via filter or manually, in an organizational role, UNIX and Tivoli Access Manager accounts are created automatically for that user.

This concludes the discussion on the integration between Tivoli Access Manager and Tivoli Identity Manager.



Part 2

Customer environment

In this part, we discuss an identity and credential management business solution for our sample Tivoli Austin Airlines corporation, which operates on a worldwide basis.

Archived

Tivoli Austin Airlines, Inc.

In this chapter, we provide an introduction to the overall structure of our sample Tivoli Austin Airlines (TAA) corporation, including their business profile, their current IT architecture and infrastructure, and their medium-term business vision and objectives.

Note: All names and references for company and other business institutions used in this chapter are fictional. Any match with a real company or institution is coincidental.

7.1 Company profile

Tivoli Austin Airlines is one of the major airlines within the continental United States. It has been in business for 12 years and now operates over 600 daily flights nationwide, with the motto *to fly anything, anywhere*.

The following sections describe:

- ▶ The geographic distribution of TAA
- ▶ The company organization
- ▶ Human resources (HR) and personnel procedures

Note: The following sections describe the company information relevant to an Tivoli Identity Manager implementation and are not intended to be a complete description of the company.

7.1.1 Geographic distribution of TAA

TAA is based in Austin, Texas, with the corporate head office and the central IT data center located near the Austin International Airport. TAA also operates the following three regional centers:

RW	Regional center west (San Francisco)
RA	Regional center Austin (Austin, within the central IT data center)
RE	Regional center east (New York)

These regional data centers service the IT needs of the region, such as LAN/help desk support and user administration. The corporate IT staff, such as systems programmers and developers, are located at the central IT data center.

TAA also runs multiple customer service centers (CSC) in the major airports, servicing front-office functions, such as ticketing, member lounges, baggage management, and staff HR systems. The CSCs have no local staff, so they contact the regional centers for technical support.

The TAA sites are:

Austin, TX	This is the IT center housing the core IT infrastructure and staff. It is also the regional center for the Austin region and contains the technical support staff for the central region. There is also a customer service center.
San Francisco, CA	This site is the regional center for the west region. It contains the technical support staff for the west region. There is also a customer service center.

New York, NY	This site is the regional center for the east region. It contains the technical support staff for the east region. There is also a customer service center.
Seattle, WA	This site contains a customer service center, is part of the west region, and is supported by the regional center in San Francisco.
Los Angeles, CA	This site contains a customer service center, is part of the west region, and is supported by the regional center in San Francisco.
Denver, CO	This site contains a customer service center, is part of the Austin region, and is supported by the regional center in Austin.
St. Louis, MO	This site contains a customer service center, is part of the Austin region, and is supported by the regional center in Austin.
Detroit, MI	This site contains a customer service center, is part of the east region, and is supported by the regional center in New York.
Raleigh, NC	This site contains a customer service center, is part of the east region, and is supported by the regional center in New York.

The geographic distribution of TAA is shown in Figure 7-1. The figure shows the three regions: west, central, and east.

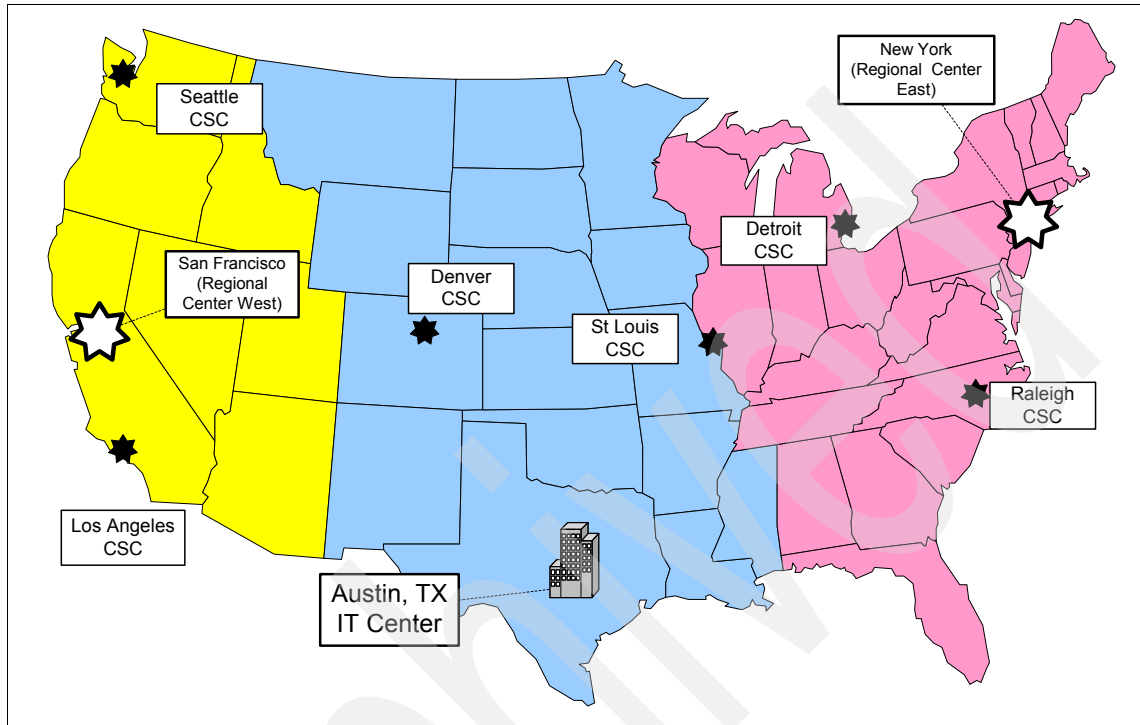


Figure 7-1 TAA geographic distribution

TAA's expansion

To fly anything, anywhere is the motto for TAA. Since this represents the business purpose of the company, TAA has just opened a new customer service center outside of the United States of America. This CSC is located in Mexico City, inside the Mexico City International Airport, and provides all kinds of services to customers from Mexico including online ticket sales and account management for the TAA frequent passenger benefits. This effort complies with the business objective of TAA becoming one of the premiere airlines in the world. Figure 7-2 depicts the location of the new CSC.

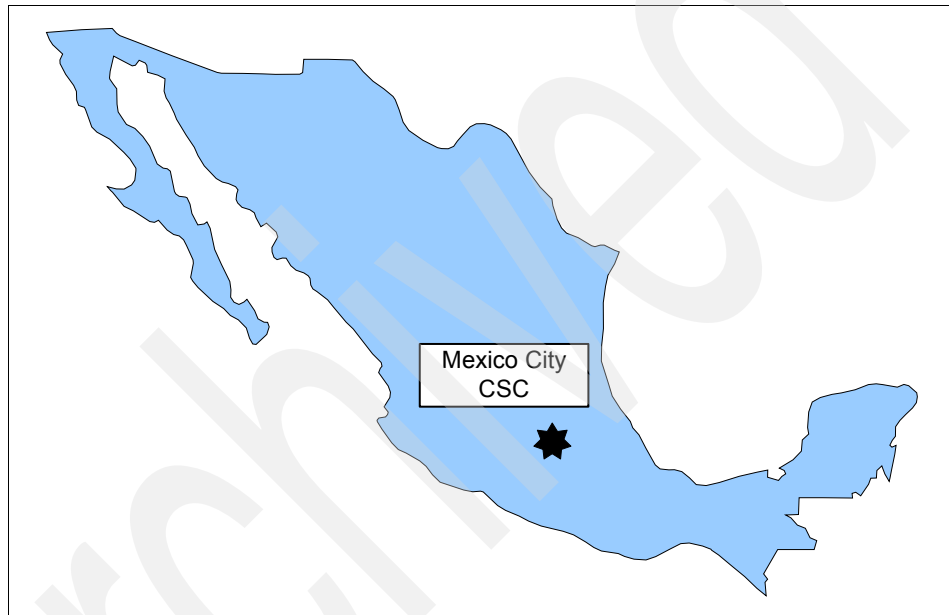


Figure 7-2 TAA expansion

The CSC in Mexico City provides the same services as the US locations, such as servicing front-office functions, ticketing, member lounges, baggage management, and staff HR systems. This CSC has no local support staff. They contact the regional center in Austin for technical support.

Since the official language in Mexico is Spanish, TAA has implemented language support on most IT systems to provide Spanish locale support to messaging services, including e-mail, alerts, and self-service Web pages.

7.1.2 Organization of TAA

The company is split into four key areas: the three regions and a core services division. This is shown in Figure 7-3.

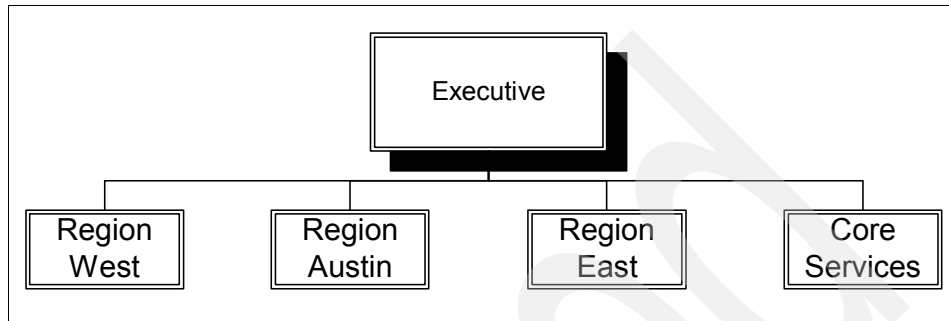


Figure 7-3 High-level organization chart

Each of the regions is responsible for the operation of the local services in that region, including customer service, baggage handling, ground services, airport liaison, and staffing. The three regions have the same structure. The organization chart for the central region is shown in Figure 7-4.

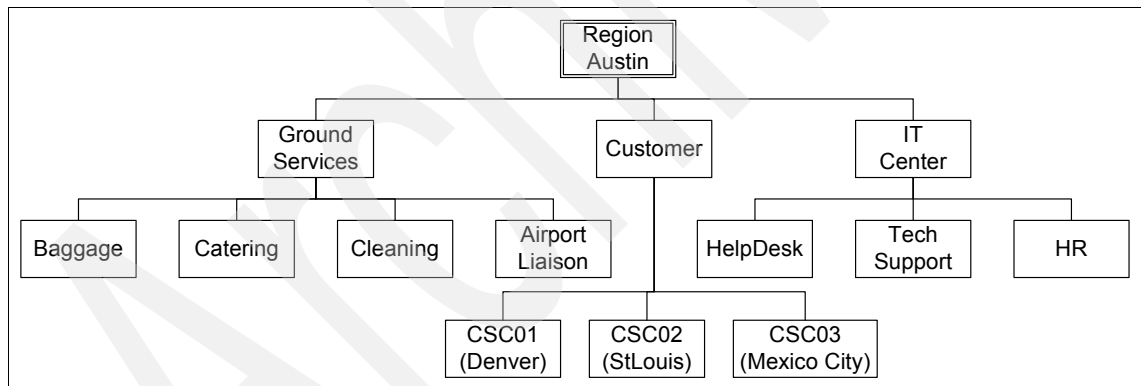


Figure 7-4 Central region organization chart

The core services division acts on a company-wide scale. It is split into three departments:

- ▶ Sales
- ▶ Support
- ▶ Flights

Each of these departments has a number of teams, as shown in Figure 7-5.

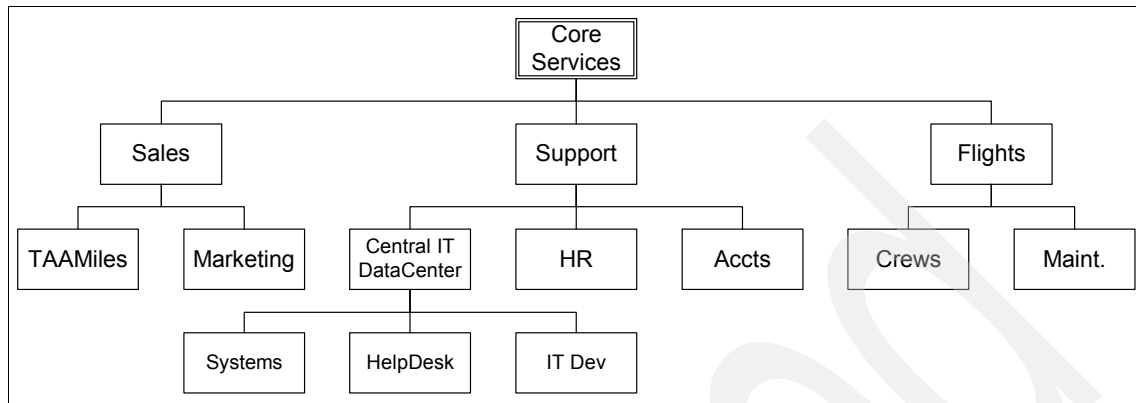


Figure 7-5 Core services organization chart

Each team within a department has a unique business code identifying the team and its location.

7.1.3 HR and personnel procedures

Personnel are managed locally within each region for that region, and by the core services HR team in Austin for all core services staff. The following procedures apply to personnel management:

- ▶ When a new employee joins the company, he is added to the HR system. An e-mail (using Lotus Notes) is sent to the new employee's manager indicating when the person is starting work and his HR details. The manager determines what types of access each person needs and sends e-mails to the appropriate support team to create accounts on the systems (for example, the LAN team creates the Windows domain account and the Linux team creates the Linux accounts). When access is granted, an e-mail is sent back to the user's e-mail account giving account details, including their password. As the support teams are small, there is often a delay of a few days in creating each account.
- ▶ When an employee needs additional access to resources, he has his manager ask the appropriate support team, via e-mail, for the access. As with new accounts, the support teams grant the additional access.

- ▶ When an employee forgets his password or has an account locked due to invalid passwords, he must call the help desk (either at the regional or central level). The help desk can reset Windows (LAN) and z/OS® RACF passwords and accounts, but Linux resets must be referred to the Linux support team.
- ▶ When employees leave the company, they are removed from HR, and sometimes their accounts are deleted. However, this is not applied consistently.

Each employee has a jobcode to describe their job role. Some of these are common across the regions, such as CSC manager and CSC administrator. Some jobs are specific to a team, such as Linux Sysadm. These job roles and jobcodes are managed by the central HR team. They rarely change.

When a new employee joins the company, there is some manual provisioning that must be performed, as follows:

- ▶ All customer-facing staff, such as ticketing agents, air crews, and ground personnel, must have uniforms ordered. This is normally carried out by the new employee's manager sending an e-mail to the uniform department, which then contacts the new employee and arranges for fitting and ordering of uniforms.
- ▶ All non-customer-facing staff, such as the administrative and IT staff, must have real estate set aside for them, including desk locations, phone connection, and filing cabinets. This is normally carried out by the new employee's manager sending an e-mail to the local office manager, who arranges everything.

7.2 Current IT architecture

In this section we describe the current IT environment at TAA. We cover:

- ▶ An overview of the TAA network
- ▶ The recently implemented e-business application
- ▶ The security infrastructure deployed for the e-business application
- ▶ The secured e-business initiative architecture
- ▶ User administration issues

7.2.1 Overview of the TAA network

TAA's central IT data center has implemented a back-end datastore, which is based on DB2 running on z/OS. They are using an MQ Series infrastructure for asynchronous transactions between the central IT data center, the CSCs, and the regional data centers.

The high-level network diagrams of TAA's network are shown in Figure 7-6.

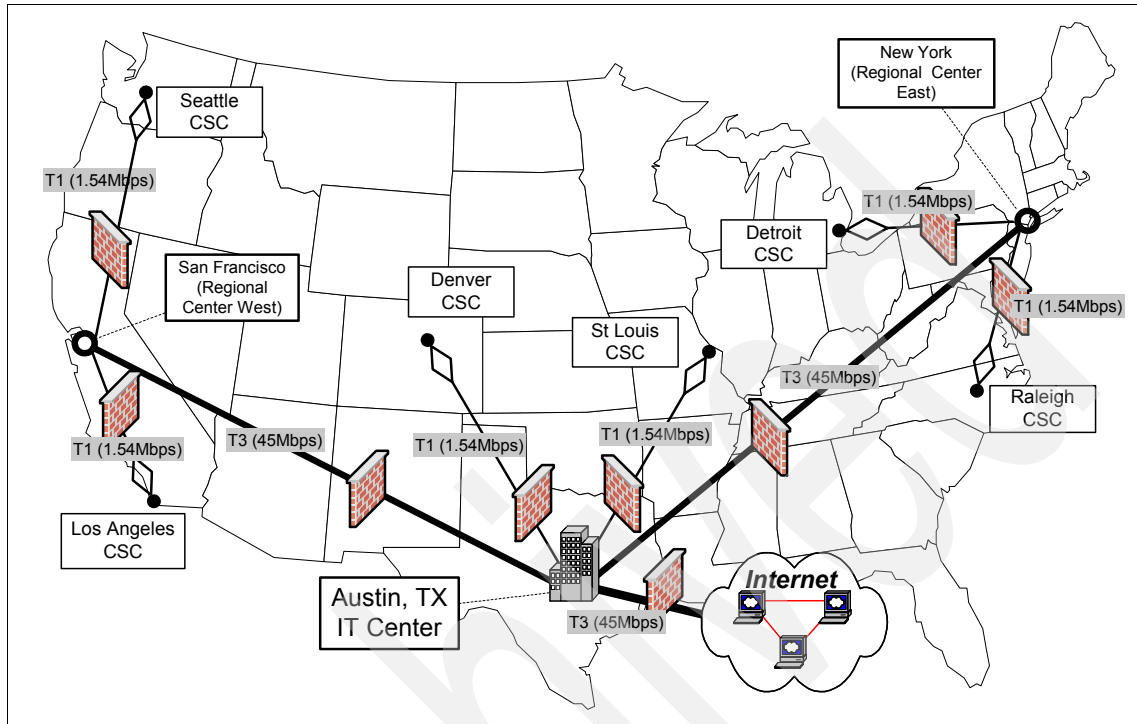


Figure 7-6 The TAA network

Figure 7-7 shows a high-level network diagram of the CSC in Mexico City.

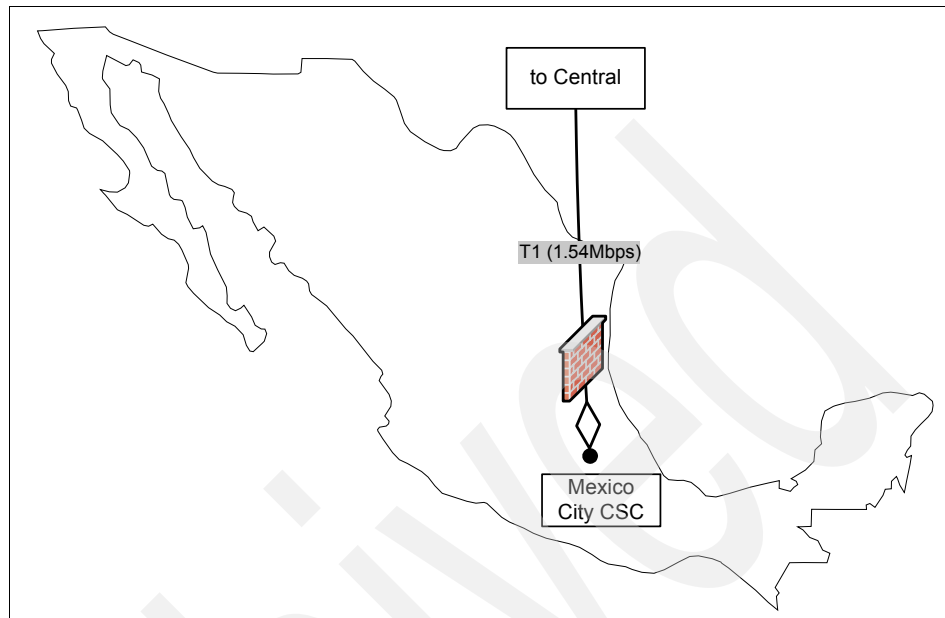


Figure 7-7 Mexico City CSC connectivity diagram

The locations of firewalls are shown in Figure 7-10 on page 313. All external access to the TAA network is channeled through the firewalls and routers in Austin. There are also firewalls between the regional centers and the IT data center, as well as between the regional centers and the CSCs.

All T1 and T3 links are leased services. TAA relies on the service provider to ensure the necessary uptime, as agreed upon in the service level agreement. The diagrams in Figure 7-6 on page 307 and Figure 7-7 are therefore logical and do not show redundant/standby links or triangulation of the network. TAA relies on the service provider for this.

TAA uses Lotus Notes as its e-mail system. This application is not available in the CSC or at the gate terminals.

7.2.2 TAA's e-business initiative

Most of the business applications have been migrated to a distributed WebSphere Application Server implementation based on Linux systems, which is located in every regional center. All these systems communicate with the back-end database through the high-speed network.

The only application that has not been implemented using the WebSphere model is the *gate terminal application*. This application runs on a Windows Active Directory-based network on Windows terminals in each CSC (that is, the CSC employees cannot use a browser to access this application). The gate terminal application uses MQ Series calls to check the appropriate passenger data. Although a risk assessment has highlighted this MQ area as being in need of some work, particularly as messages can sit on queues in an unencrypted form, TAA decided that other risks were of a higher priority.

The implementation of Tivoli Access Manager for e-business was important for phase one of improving security, but it also provides a platform for the future authorization services, specifically for MQ series, using Tivoli Access Manager for Business Integration.

The identity management solution therefore must be able to provision to operating systems (for native MQ), Tivoli Access Manager, and the e-mail system based on Lotus Notes.

7.2.3 Security infrastructure for the e-business initiative

In the past, TAA's application system experienced many unauthorized access attempts to critical business data. Recently, TAA has deployed a security solution that implements a centralized access control mechanism enforcing authentication and authorization of users before they access the applications and critical data via their Web browser. This solution is implemented based on Tivoli Access Manager for e-business, with the access control component being WebSEAL.

Note: In this IBM Redbooks publication we omit any detailed description of the Tivoli Access Manager and WebSEAL solution because our focus is on the identity management system. For further details consult the following IBM Redbooks publications:

- ▶ *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014
- ▶ *Enterprise Business Portals with IBM Tivoli Access Manager*, SG24-6556
- ▶ *Enterprise Business Portals II with IBM Tivoli Access Manager*, SG24-6885
- ▶ *Deployment Guide Series: IBM Tivoli Access Manager for e-business V6.0*, SG24-7207

A typical user access with WebSEAL controls looks like this:

1. A user in a CSC logs on to the Windows domain by specifying his Windows user ID and password.
2. He starts his Web browser and accesses a login page for a specific application. He logs in with his application user ID and password. A credential is used for access control by WebSEAL in the regional centers to which the CSC belongs.
3. WebSEAL accepts or denies the login. WebSEAL works as a reverse proxy between the user's Web browser and the application-hosting Web server, controlling whether a user can access the requested resource.
4. WebSEAL's access control decisions are based on the information held within the Tivoli Access Manager Policy Server and the relevant Lightweight Directory Access Protocol (LDAP) repository. The policy server stores the access control information used by WebSEAL and distributes access control information database replicas to all defined WebSEAL servers, while the LDAP server has the user credential information created and used by Tivoli Access Manager. The policy server is located in the Austin site, but WebSEAL and LDAP replica servers are made available in each regional center. The LDAP server in Austin is the master server, which can be modified. The LDAP replica servers are read-only.

Only the Web applications can be secured by WebSEAL using Web user accounts, but there are other types of accounts necessary to run standard operations, such as Windows, Linux, and z/OS. These accounts can only rely on the native operating system security. That is why TAA puts the employees under an obligation to follow additional security policies to strengthen the levels of security, such as a periodical password change and other password policies for all types of accounts.

At the time of implementing this security solution, TAA has started to provide new Web-based customer services (customer's mileage information, special campaign information, and so on) via the Internet. A customer's access to his data is controlled by WebSEAL also.

7.2.4 Secured e-business initiative architecture

Figure 7-8 depicts only the Austin site, which consists of the central IT data center, Region Center Austin, and CSC, in order to make it simple. While there are strong grounds for altering the network topology and firewall configuration so that the Austin regional site is separate from the Austin corporate site, the risk assessment carried out once again showed that there were higher priorities (those addressed by Access Control and identity management) than this internal network topology change.

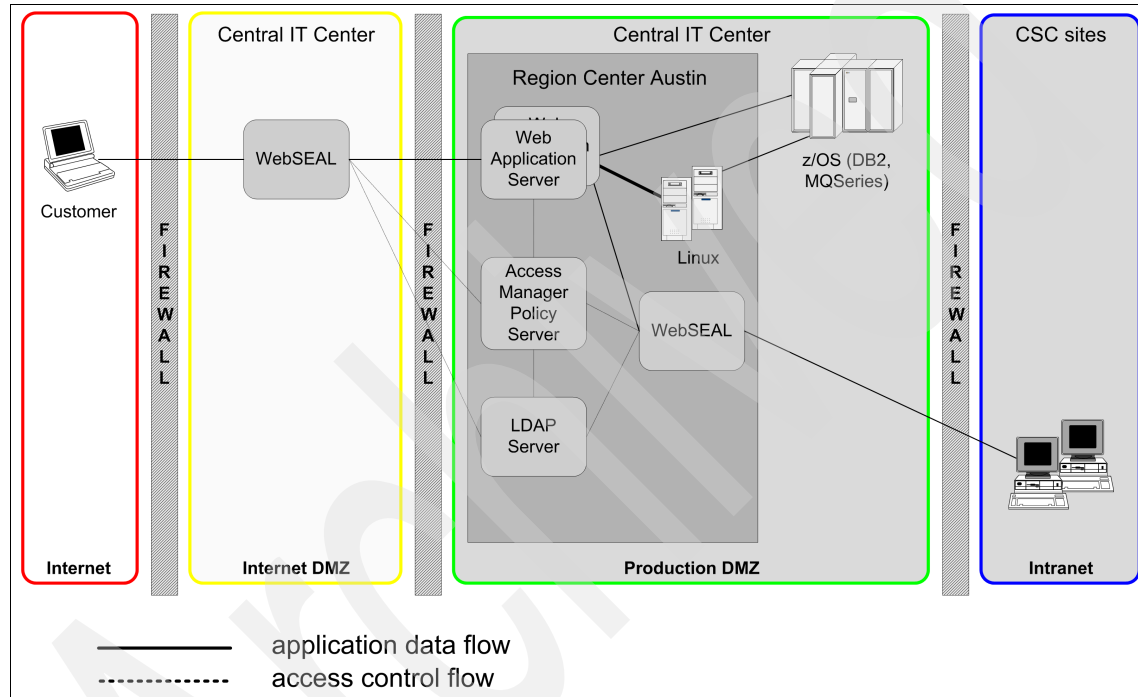


Figure 7-8 Current IT architecture in Austin

The full existing TAA topology is shown in Figure 7-9.

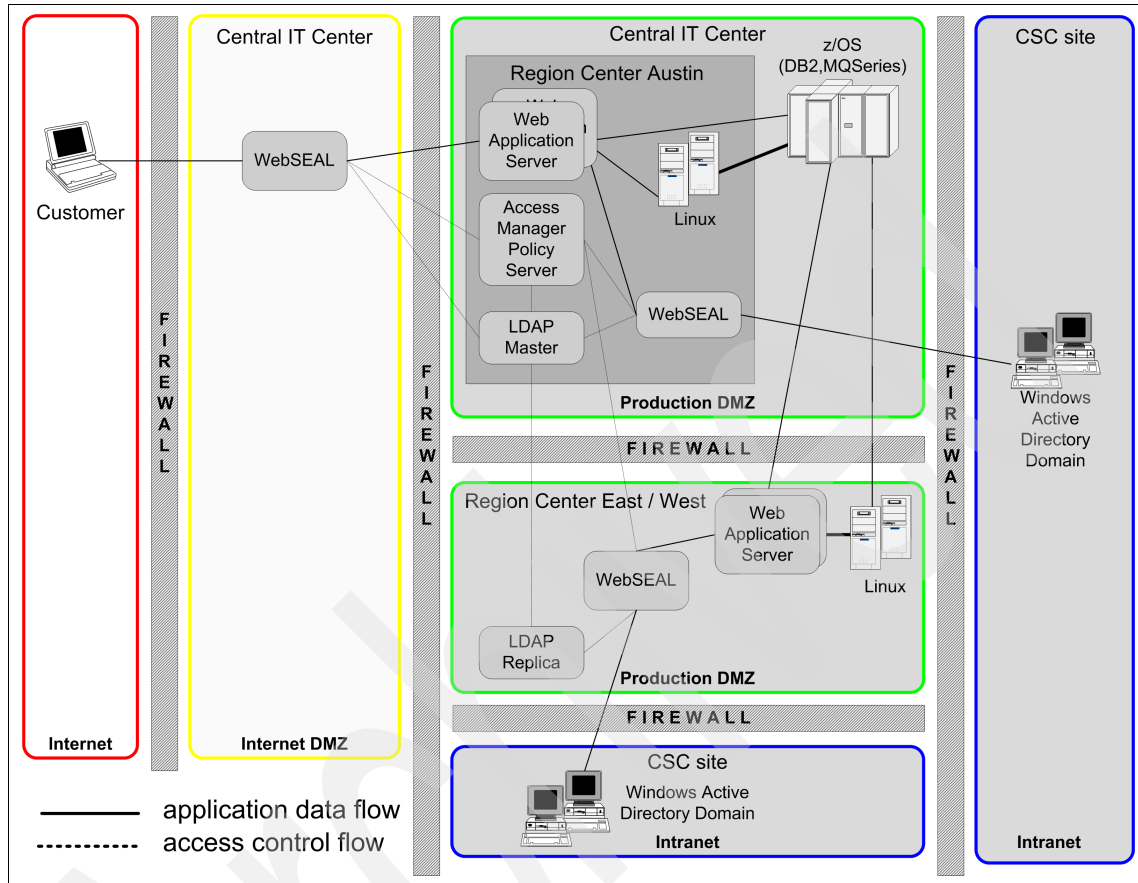


Figure 7-9 Current entire TAA architecture

Ultimately, TAA will aim at segregating the data (DB2 on zOS) and systems management zones (LDAP, security management, and so on) from the Austin corporate network. The Austin regional network could then further be separated by a firewall, and be treated exactly as though it were a remote regional center, even though it is not remote from Austin. Evolving towards this security best practice also creates further operating gains on the system management side of the organization and therefore user experience. One possible future for TAA is shown in Figure 7-10.

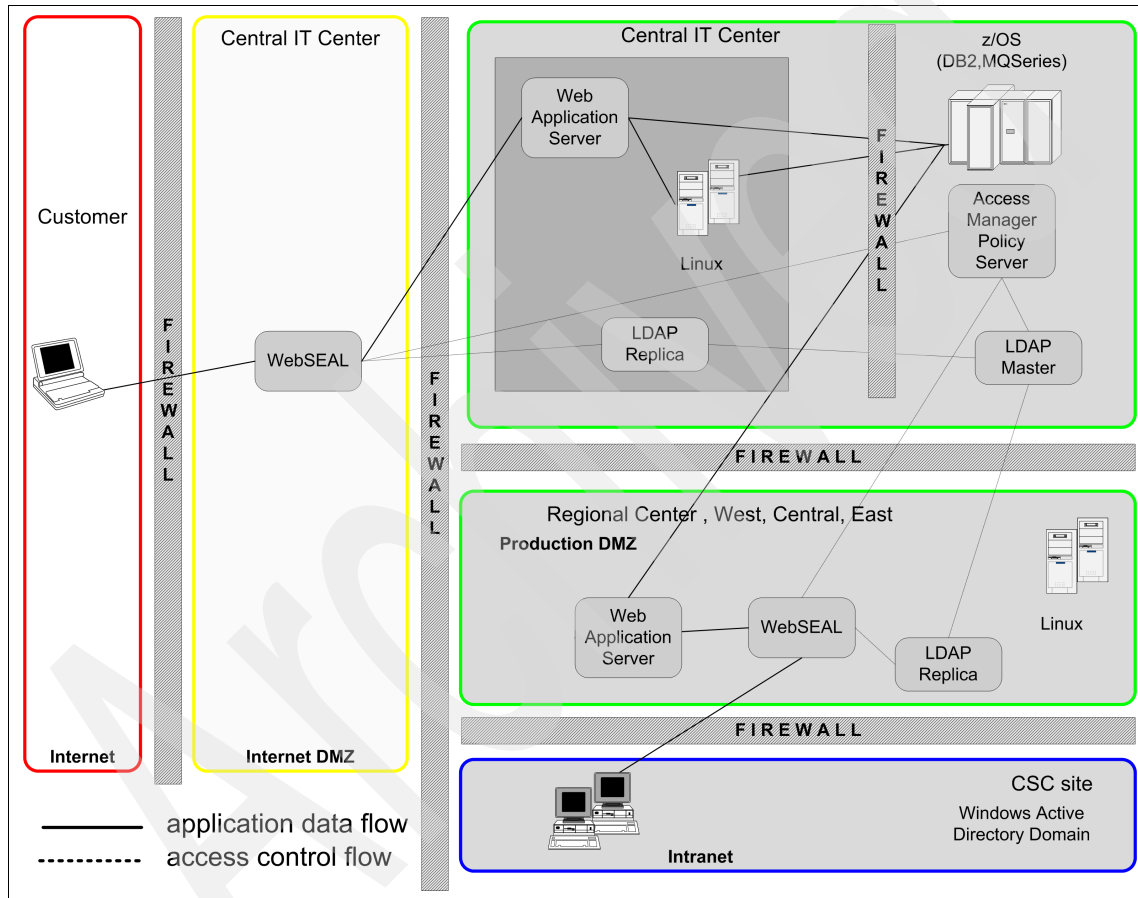


Figure 7-10 Possible future TAA architecture

Undoubtedly, the security solution shown in Figure 7-9 on page 312 contributes to strengthening the corporate security level and customer confidence in using the Internet over previous architectures. The secure implementation helps build and maintain a positive company image, but there are other emerging problems, particularly in the area of identity management.

7.2.5 Identity management and emerging problems

Emerging problems are related to user administration and identity management. Before we describe the emerging problems in detail, we provide an overview of the current user management.

Current user management

TAA uses many different platforms with user accounts in their system. The CSC staff have their accounts in a Windows domain and the Tivoli Access Manager LDAP directory (for WebSEAL-controlled Web application access). The regional center administrators have their accounts on Linux systems, adding to Windows domains in their region and the Tivoli Access Manager LDAP directory. All these accounts are managed by the regional center administrators using OS/application-specific interfaces. Windows domain accounts are created with the Windows Administration Console, Linux accounts with the relevant administration interface for the release, and Tivoli Access Manager accounts with the pdadmin utility shipped with Tivoli Access Manager for e-business.

Programmers and developers in the central IT data center have their accounts on all platforms in the enterprise, including z/OS. Due to their specific tasks and influence, they often manage their own accounts with administrative rights, not the regional administrators. Accounts on z/OS are created using the RACF interface shipped with z/OS.

Customers' accounts are created in the Tivoli Access Manager LDAP registry located in Austin. They are managed by administrators in the regional center Austin.

Emerging problems

As mentioned above, all employees have several types of accounts and they must maintain multiple sets of user IDs and passwords. If an employee forgets his password, he must call a regional administrator to reset the password. Especially after the new periodical password change policy has been applied, more and more password reset requests come to the regional administrators. The current password reset flow is:

1. A user who wants to reset his password must use a hardcopy request form specifying the type of account, account name, and reason for the reset request.
2. His manager accepts the request form and signs for approval. In the case of denial, the request form is returned to the requester.

3. The user faxes the form to his regional center after he gets his manager's approval.
4. An administrator resets the specific accounts and sends an e-mail to the user and his manager to inform them of the password reset and the new temporary passwords.

Too many requests cause regional administrators not to reply immediately, and some employees cannot continue their work because their accounts remain inaccessible. When a user's manager is not available temporarily, the password reset procedure is delayed even more.

Besides password reset requests, there are other reasons that the administrator's workload increases. Employee fluctuation is a much more common scenario these days, which brings newly hired people, laid off employees, staff changes, and so on. This increases the administrator's workload even after security solutions have been deployed and account types have been added.

Furthermore, the management interfaces for the various types of accounts on the different platforms are not the same. Administrators must use specific interfaces along with account types (for example, pdadmin for Tivoli Access Manager accounts). There are many complex operations and much more time is needed for an administrator to learn how to properly use them. In complex operations, human error is a common problem.

A similar situation prevails in the administration of customer information: requests for creating their accounts, password reset requests, and so on. Some customers have already contacted IT management because it takes too much time and effort to manage their account information.

As we have seen, complicated operations prevent regional administrators from working functionally. This decreases employee and customer satisfaction, and it leads to an inability to provide sufficient services to the users in a proper way.

TAA has now decided to implement an additional security management solution focusing on user and identity management. The main objective in this case study scenario is to use the Tivoli Identity Manager solution.

Because of the implementation of the Sarbanes-Oxley Act of 2002, many companies are trying to deal with this issue. TAA is aware they must comply with major historical standards in order to be a company that its customers can trust (which is a very important point for e-business initiatives), and TAA is now working in all areas involving historical matters, such as those discussed in Chapter 1, "Business context for identity and credential management" on page 3.

7.3 Corporate business vision and objectives

TAA has implemented its e-business application system to employees and expanded its services on the Web to its customers. This system relies on a Web-based application infrastructure provided by the IBM WebSphere Application Server and a centralized access control solution using Tivoli Access Manager for e-business.

In order to increase employee productivity and prevent dissatisfied customers, the user management processes for all the involved platforms must be streamlined and opportunities for human errors must be reduced, if not eliminated.

The TAA mid-term vision is:

- ▶ TAA wants to deploy a corporate-wide user management system to be operated efficiently and correctly, following the corporate security policies. It needs generalized information about user management to make plans for the future to adjust the system to its change in circumstance. Also, in order to lessen administrative cost, it is desirable to automate management operations wherever possible.
- ▶ TAA already has some systems management functions in place, such as monitoring system availability and software distribution, which are implemented based on the Tivoli Framework. An additional user and identity management system should be implemented with minimum development cost, making full use of the existing resources.

7.4 Project layout and implementation phases

Based on the corporate business vision, TAA has decided to implement the new solution in four phases:

1. The first phase concentrates on the preparation for tasks that must be implemented before Tivoli Identity Manager can be committed to production. These tasks include the system installation and verification of the correct operation of the components. This initial phase also creates the HR feed process and runs the first reconciliations and orphan account cleanup. For more details see Chapter 9, “Technical implementation: Phase I” on page 341.
2. Phase 2 focuses on functionalities that do not require giving people access to Tivoli Identity Manager (creation of Tivoli Identity Manager accounts is not necessary) or performing mass training.

Common accounts are created for new hires, and accounts are suspended on termination. In addition to these account-related tasks, the password synchronization function using the Windows password interceptor is going to be implemented. For more information refer to Chapter 10, “Technical implementation: Phase II” on page 419.

3. Phase 3 implements all other required functionality except RBAC and separation of duty policies (SoD). This includes the challenge/response for forgotten passwords, account maintenance through the Tivoli Identity Manager Web user interface, delegated administration, which will include group management, and approval workflows. Finally, regional accounts are automatically granted/suspended based on transfer in the HR feed and compliance alerts are generated. For more details see Chapter 11, “Technical implementation: Phase III” on page 449.
4. The final phase, 4, focuses on the full RBAC enablement and defines organization-wide roles, identifies role relationships, and then build role hierarchies, and provisioning policies for those roles. In addition to defining roles and provisioning policies, a self-service interface is provided in order to request role changes. To further strengthen compliance with the Sarbanes-Oxley Act of 2002, roles that have conflicting interests will be managed by Identity Manager separation of duty policies. For more details see Chapter 12, “Technical implementation: Phase IV” on page 511.

7.5 Return on investment (ROI) study and results

When considering the implementation of a centralized identity management solution, there are two business drivers that are important:

- ▶ Does the deployment of the tool mitigate security risk to an extent where the residual risk is acceptable to the business?
- ▶ Does the return on investment of the proposed solution occur in an acceptable time frame?

TAA, therefore, asked for an ROI study to be conducted to assist it with producing a business case to present to senior management.

Note: If you wish to conduct an ROI study for your particular situation, a comprehensive ROI questionnaire tool is available from an IBM Tivoli Sales representative.

In addition to the company profile and information discussed above, TAA supplied information concerning operational costs, which was factored into the ROI case along with the costs of software, additional hardware, and deployment services. When information was not available, the default values supplied by the ROI tool were used. The tool included some elements of risk analysis and also looked at the benefits from integration with the existing authentication and authorization framework (Tivoli Access Manager for e-business) that was already in place.

The decision to move ahead into the implementation phase was made not only as a result of the current ROI benefits, but also because of the benefits to future business-led projects that would no longer need to code complex security models. This means a more rapid deployment of the application (product or service) and therefore the ability to set a higher margin price, because as the first entrant to the market, there would, for a period, be no competition.

The results of the ROI study conducted for TAA amounted to a detailed 80-page report. The executive summary and some other parts of the report are shown in the following sections.

Executive summary

Tivoli proposes the implementation of a business impact management solution to help the organization achieve its strategic goals. The implementation of this solution requires an investment of a \$1,863,830 total 3-year investment.

As a result of this investment, the organization is expected to realize savings and business benefits of \$870,330 over the next 12 months, and \$4,460,962 total over the 3-year analysis period. Comparing these benefits to the investment, the recommended solution has a return on investment of 139%, and net present value (NPV) savings of \$1,986,406. The initial investment is recouped with a payback period of 15 months.

Strategic business initiatives

The analysis of the company's business needs uncovered the following strategic business initiatives:

- ▶ Risk management

Goal 1: Reduce security risks by ensuring the application of corporate security policies concerning identity management.

Goal 2: Reduce risks and build customer confidence by using Tivoli Identity Manager features to strengthen compliance with standards, such as the Sarbanes-Oxley Act of 2002.

- ▶ Quality of service
 - Goal 1: Reduce the waiting time for password resets for users, thus improving user satisfaction.
 - Goal 2: Integrate the non-IT provisioning system (uniform, desks, phones, and so on) with the IT user account provisioning system to ensure better responsiveness.
- ▶ IT best practices
 - Goal 1: An administrator should have access to only the resources needed, not to all systems.
 - Goal 2: An administrator may be able to self administer his accounts, but should do so through a centrally fully audited system.
 - Goal 3: Local resource changes should be detected and be subject to the same security controls as those carried out directly on the centralized identity management system.
 - Goal 4: Implement technical controls to detect and manage conflicts of interest.
- ▶ Strategic advantage
 - Goal 1: TAA must become customer Web enabled. This includes the use of automated identity management that enables new business initiatives to be integrated without new identity solutions being required.
 - Goal 2: Costs of customer identity management must be driven down in order to reduce overhead and improve margins.

To address these strategic business initiatives, the following Tivoli Solutions were selected:

- ▶ Tivoli Identity Manager
- ▶ Tivoli Access Manager for e-business (already in use)

ROI analysis

Table 7-1 compares the investment costs for the Tivoli solution with the savings and business benefits. Financial analysis examines the projected cash flow over three years to calculate ROI, NPV savings, and payback period.

Table 7-1 ROI analysis

ROI analysis	Initial	Year 1	Year 2	Year 3	Total
Total adjusted costs	\$528,000	\$606,038	\$359,280	\$370,512	\$1,863,830
Total adjusted benefits	\$0	\$870,330	\$1,659,405	\$1,931,227	\$4,460,962
Cumulative adjusted costs	\$528,000	\$1,134,038	\$1,493,318	\$1,863,830	
Cumulative adjusted benefits	\$0	\$870,330	\$2,529,735	\$4,460,962	
Net benefit	(\$528,000)	\$264,292	\$1,300,125	\$1,560,715	\$2,597,132
Cumulative net benefit	(\$528,000)	(\$263,708)	\$1,036,417	\$2,597,132	
Three-year net savings	\$2,597,132				
Net present value (NPV)	\$1,986,406				
Internal rate of return (IRR)	121%				
Return on investment (ROI)	139%				
Payback period (break-even)	15 months				

ROI analysis graph

The ROI analysis graph shown in Figure 7-11 illustrates the cumulative investment cost versus the cumulative savings and business benefits over the 3-year analysis period.

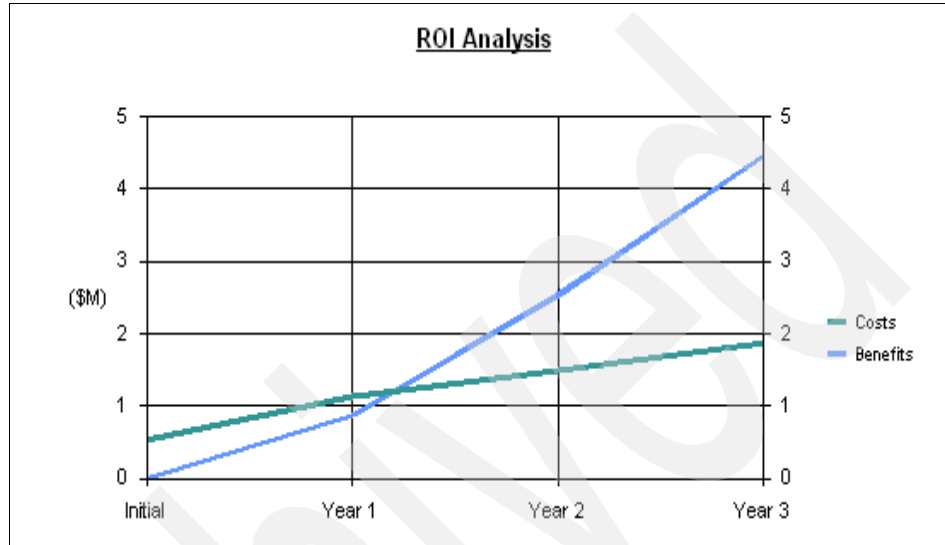


Figure 7-11 ROI analysis graph showing 15month cross-over point

Important: Other factors taken into consideration for this ROI study were a simplified risk analysis and financial measures, such as internal rate of return. The cost of the software selected was calculated using the current IBM Passport Advantage® rules for the United States. As with any Passport Advantage Pricing figures, they are dependant upon a number of factors, including an organization's current discount band. The software pricing for TAA, therefore, should not be viewed as applicable to your organization. For a full ROI study, including a current pricing assessment using up-to-date figures and metrics, contact your local IBM Tivoli Account Manager.

Archived

Identity management design

In this chapter we describe the business requirements, functional requirements, security design objectives, and design aspects for an identity management foundation based on Tivoli Identity Manager.

Most implementations are done in multiple phases. Tivoli Austin Airlines (TAA) has decided to use a multi-phased approach so as to gain some return on investment (ROI) advantage early in the project. The content of each phase is decided by analyzing the priorities of the business requirements and mapping these through their functional requirements to Identity Manager capabilities. The earlier phases are dedicated to satisfying those requirements associated with high-priority business requirements and low-cost Identity Manager capabilities.

Implementation details for each of the phases are found in 7.4, “Project layout and implementation phases” on page 316.

8.1 Business requirements

Tivoli Austin Airlines Inc. has implemented its e-business system with Web-based technology and a Tivoli Access Manager based access control architecture. Its has discovered that there are emerging problems, as described in Chapter 7, “Tivoli Austin Airlines, Inc.” on page 299.

From the vision and objectives presented in 7.3, “Corporate business vision and objectives” on page 316, the CEO emphasizes the following eight business requirements for the project.

- ▶ All administrative operations related to user and account management, including creation, modification, suspension, and password reset, must be executed correctly and in a timely manner.
 - Operations that do not require approvals should execute momentarily.
 - Operations that require approvals must not languish waiting for a response. Stalled requests must be escalated. It must be possible to share the burden of approvals.
 - It must be possible to measure the timeliness of completion of account management requests so that this can be measured against TAA’s service level agreements.
- ▶ Reduce the costs of administering users and their accounts. The CEO is keen to gain cost savings by reducing the amount of work that the administrators must do. The areas identified where savings could be made include:
 - The effort required to reset passwords for users who have forgotten theirs
 - The effort required to manually create accounts when a person joins the company
 - The effort required to add new accounts (and remove old accounts) when an employee changes job roles
- ▶ The corporate security policy should be enforced for all user accounts and their attributes, access rights, and password rules. User accounts inconsistent with the policy should generally not be allowed.
- ▶ The identity management solution must not be so rigid that it prevents TAA from responding to emergencies and temporary exceptional needs. Administrators must be able to override the system’s defaults and policies when necessary.
- ▶ The user and account management historical data must be available from a corporate-wide perspective in order to verify whether the system works according to the guidelines and policies. These logs can help the company understand shortcomings and implement future improvements.

- ▶ Improve audit compliance. TAA recently had an external audit performed and a number of areas were seen to be lacking:
 - Many employees have access to systems that they should not because they:
 - Changed job roles and retained access from their old job role.
 - They are friends with the system administrators and were granted special access without any form of independent check or review of the request.
 - They have left the company, but their accounts have not been deleted.
 - There is no reporting available to verify security compliance.
 - There is no periodic certification of users' access rights.
- ▶ The identity management solution must be implemented in a secure manner. It must ensure that:
 - Sensitive data is protected from unauthorized access.
 - Audit data is protected from unauthorized alteration.
 - The system is protected from unauthorized users.
- ▶ The identity management solution must be bilingual. Its user interface, reports, and e-mail notifications and documentation must be available in both English and Spanish.

8.2 Functional requirements

We extract functional requirements by mapping business requirements to their underlying reasons. We expand the reasons in increasing detail until we find problems that can be solved using capabilities of Tivoli Identity Manager. Our functional requirements tie these low-level reasons for a business requirement to the Identity Manager capability that will fulfill that business requirement.

Let us examine each business requirement and search for reasons and the functional requirements.

- ▶ Business requirement 1: Identity management should be executed quickly and correctly.

There are two main problems in this area: System administrators are unable to keep up with the volume of requests and approvals are not being processed in a timely manner.

The biggest burden on administrators is the increasing number of password reset requests. After implementing a central security solution for access control and applying a new security policy for passwords, users must change passwords more frequently than before. This leads to users forgetting their passwords more often, which results in many password reset requests. Users are less likely to forget their passwords if they use the same password for all of their accounts. If they do forget their password, we can reduce the burden on system administrators by delegating the ability to do password resets. This may be done by users' managers or possibly by the users themselves. This leads to the first two functional requirements shown in Table 8-1.

Table 8-1 Functional requirements for timely password management

Requirement	Description
A	Users will have a single password for all of their accounts.
B	Password resets will be delegated to users other than the system administrators (possibly to the users).

Another reason that system administrators have trouble keeping up with the rate of requests is that user and account management operations are time consuming and skill intensive. Administrators must waste time manually entering data that could be calculated automatically. This is not only time consuming, it is also error prone. This leads to administrators taking more time to repeat requests that were done incorrectly.

Administrators must also learn different management interfaces for each type of account. Administrative productivity could be enhanced by utilizing a common interface to manage different types of accounts centrally.

This leads to the next set of functional requirements, shown in Table 8-2.

Table 8-2 Functional requirements for timely account management

Requirement	Description
C	Common values are entered automatically.
D	Manually entered values can be checked for correctness.
E	Provide a common user interface for administration.

The other major cause of delays in user and account management is the request approval process. TAA has identified three primary causes for delays in granting approvals:

- An approver may not be available at the time of a request. Requests should not be delayed because an approver is out of the office. Approvers should be able to delegate their responsibilities if they know that they will be unavailable.
- Approvers may be too busy or receive too many requests to respond quickly. Approvals should be assigned to teams instead of to individuals. It must be possible for the team members to assign and take ownership of individual approval requests.
- Approvers may forget that they are responsible for a request. An approver who does not act on a request must be periodically reminded that the request is waiting. If he still does not respond, the request should be escalated to a different approver.

These issues are addressed by the next set of functional requirements, shown in Table 8-3.

Table 8-3 Functional requirements for timely request approval

Requirement	Description
F	Allow delegation of approval responsibilities.
G	Support collaboration by multiple approvers.
H	Remind approvers of waiting requests.
I	Escalate ignored requests.

- ▶ Business requirement 2: Reduce administrative costs.

TAA has identified three areas in which they want to reduce the costs associated with user and account administration:

- Password resets
- Account creation for new employees
- Account maintenance for users who change job roles
- Distributed group management

These four tasks occupy much of the time of many highly paid system administrators. We can reduce the number of administrators and allow the remaining administrators to focus on higher value projects if these tasks can be automated or delegated to other users.

Password resets have already been discussed in the context of business requirement 1 (execute requests quickly and correctly). Functional requirement B (delegation of password resets) also satisfies the cost reduction business requirement.

TAA's administrators are responsible for managing groups over all main platforms and applications (such as Microsoft Active Directory and Linux servers). They are also responsible for creating new accounts for newly hired employees. Some of these accounts, such as e-mail and Windows access, are common to all employees and use similar settings on all accounts. The automation of the setup of these accounts would allow system administrators to concentrate on more useful work.

System administrators are also responsible for creating new accounts and suspending existing accounts when employees change job roles. Many job roles have a standard set of accounts and access rights that must be given to a user when he enters the role, and must be removed when the user leaves the role. This is another case where automation could relieve the administrators from repetitive tasks.

The functional requirements for cost reduction are shown in Table 8-4.

Table 8-4 Functional requirements for cost reduction

Requirement	Description
B	Password resets will be delegated to users other than the system administrators, possibly to the users.
J	Automatically create common accounts when a person is employed.
K	Automatically add and remove accounts and access rights when a user changes job roles.
DD	Centralized group management.

- ▶ **Business requirement 3:** The corporate security policy should be enforced for all user accounts.

Accounts sometimes have attributes that do not comply with the corporate security policy. This may be accidental due to mistakes made by administrators or ignorance of the violated policies. Some non-compliant accounts may also be the result of intentional misconduct by administrators. These may be cases of administrators who are too lazy to follow the policy, or the administrators may have malicious reasons for violating the policies. In either case, there is no verification of the values entered by system administrators when they are creating and modifying accounts.

Violations of the corporate security policies can be reduced by setting the values of account attributes automatically, when possible. Further reductions in violations can be achieved by introducing compliance checking on attribute values that are set manually. Both of these strategies rely on having a centralized user interface for account management and a way to find changes made to accounts outside of the central user interface.

There is substantial overlap between the functional requirements for insuring compliance with security policies and the functional requirements for timely account management (shown in Table 8-2 on page 326). The requirements for a common user interface, automatic calculation of common account attribute values, and checking of manually entered values all help to make system administrators more productive and to enforce compliance with the security policies. In addition, there are requirements that the security policies will still be enforced even if an account is changed outside of the centralized account management tool, and if the policies themselves are changed. The combined functional requirements for compliance with security policies are shown in Table 8-5.

Table 8-5 Functional requirements for compliance with security policies

Requirement	Description
C	Common values are entered automatically.
D	Manually entered values can be checked for correctness.
E	Provide a common user interface for administration.
L	Account changes made outside of the common interface are detected and checked against the security policies.
M	Changes to security policies are checked against existing accounts.

- ▶ Business requirement 4: Enforcement of security policies must be flexible enough to allow for emergencies and exceptions.

TAA realizes that there will always be cases where an exception to a security policy will be needed. No set of policies will ever be able to foresee every combination of account attributes that might be needed by a user. When temporary or emergency needs arise, there must be a way that the administrators can override the security policies.

TAA anticipates three likely scenarios where exceptions to the security policies will be needed. The first is when users need temporary administrative rights in order to perform software installation or maintenance. The second situation is when users change their job role. The security policy may require that persons lose some access rights when they leave their old job role. But such changes in responsibilities are rarely instantaneous. Users who are changing departments often go through a transition period during which they need the access rights of both their new and old job roles. It must be possible to detect accounts that are out of policy, and have a designated administrator define how long the account may remain out of policy before it is brought into compliance automatically. The third situation will be when a separation of duty policy detects a violation that must be allowed an exemption for a temporary period of time.

The functional requirements for flexibility in security policy enforcement are listed in Table 8-6.

Table 8-6 Functional requirements for flexible compliance with security policies

Requirement	Description
N	An administrator can create or change an account even if the resulting account violates the corporate security policies.
O	Designated administrators will be notified when non-compliant accounts are detected.
P	The designated administrators can decide how long the account may remain non-compliant. After this period expires the account will be automatically brought into compliance with the security policies.

- ▶ Business requirement 5: User and account management historical data must be available for verification and future improvements.

In the current system, account information is scattered all over the corporate systems. It is not easy to understand how many user accounts are being used in the enterprise, at what rate they are growing, and when the system should be expanded due to increasing account numbers, and so on. The information is indispensable for verifying the current system and for making future plans to expand it. A central logging system can provide this information. This requirement is shown in Table 8-7.

Table 8-7 Functional requirements for availability of historical data

Requirement	Description
Q	A central logging system is needed.

► Business requirement 6: Improve audit compliance.

TAA wants to improve its audit compliance in four areas:

- Requiring users or their managers to periodically certify the users' continuing need for their accounts and access rights.
- Removal of accounts or access rights that are no longer needed. This may be divided into three different populations of accounts:
 - Accounts belonging to users who have left the company
 - Accounts belonging to users who have changed job roles
 - Accounts that were not certified as still needed
- Reporting capabilities for finding accounts that are in violation of the corporate security policies.
- Definition of separation of duty policies, to detect and manage roles that pose conflicts of interest.

Requiring certification of need for accesses is the best way to prevent temporary accesses from becoming forgotten accesses. TAA is concerned that users who are given temporary access to an application or data will keep that access even when the access is no longer needed. It is reasonable for people to do this if they are not certain that they are finished with their work that requires the access. The problem is that people eventually forget that they have the access, and never request that it be removed. At worst, their unused accounts or access rights are left for hackers to find. At best, determining who had access to some data or an application becomes more difficult.

Removing obsolete accounts and access rights has obvious benefits for audit compliance. The functional requirements for this area will have some overlap with the functional requirements for flexible security policy enforcement, as shown in Table 8-6 on page 330. The functional requirements that administrators be notified of non-compliant accounts and that the accounts be brought into compliance at some point in time help to satisfy both the policy enforcement and audit compliance business requirements. This meets the need to remove accounts and access rights that result from a user changing job roles.

Removing accounts belonging to people who leave the company requires that Identity Manager receives regular updates from one or more authoritative sources of identity data. This data must be updated in a timely manner so that Identity Manager can disable the accounts of former employees without excessive delays.

It is possible that persons could potentially be assigned privileges that can expose the business to conflicts of interest. TAA would like the Identity Manager system to handle these conflicts. When separation of duty policies are defined, the system needs to detect and alert to violations. However, it must also be possible to grant exemptions to any violation when temporary or emergency needs arise.

The functional requirements for improved audit compliance are listed in Table 8-8.

Table 8-8 Functional requirements for improved audit compliance

Requirement	Description
O	Designated administrators will be notified when non-compliant accounts are detected.
P	The designated administrators can decide how long the account may remain non-compliant. After this period expires the account will be automatically brought into compliance with the security policies.
R	Account owners or their managers will be periodically asked to certify their continuing need for their accounts and access rights.
S	Accounts and access rights that are not certified will be disabled or removed.
T	A regular feed of identity data from authoritative TAA sources into Identity Manager will be established.
U	An employee's accounts will be disabled or removed when the identity feed shows that an employee has become inactive.
V	A reporting mechanism will be available that identifies accounts that are not in compliance with the corporate security policies.
CC	Based on a defined policy, manage roles that have conflicting interests.

- ▶ Business requirement 7: The identity management solution must be secure. A poorly designed identity management solution poses a security risk. There are three primary areas of concern:
 - Confidentiality of sensitive data
 - Identity Manager stores sensitive data in its data stores. It also transmits sensitive data between its individual components. The stored data and the data in transit must be protected from unauthorized access.

– Integrity of audit data

Identity Manager administrators have a great deal of power. By manipulating provisioning policies they could create accounts with almost any rights that they want on any platform controlled by Identity Manager. Since it is difficult to prevent an administrator from abusing his powers, it is important that an audit trail be maintained of the administrator's actions. The administrators who are being monitored with this audit data must not have the ability to manipulate the audit data.

– Authentication of system users and components

Identity Manager must be protected from access by unauthenticated or unauthorized users. Each Identity Manager component must also authenticate the other components with which it communicates.

The functional requirements for the security of the identity management solution are provided in Table 8-9.

Table 8-9 Functional requirements for application security

Requirement	Description
W	Stored sensitive data will be protected from unauthorized access.
X	Transmitted sensitive data will be protected from unauthorized access.
Y	The actions of Identity Manager users and administrators will be tracked in an audit trail.
Z	Identity Manager administrators will not be able to manipulate the audit data or settings.
AA	Identity Manager components will be protected from access by unauthenticated or unauthorized users.

- ▶ Business requirement 8: The identity management solution must support English and Spanish speaking users.

TAA wants the employees of the Mexico offices to be able to access the identity management solution in their native language. This is important so that these employees will understand the actions that they are performing with the system. The functional requirement for this area is provided in Table 8-10.

Table 8-10 Functional requirement for national language support

Requirement	Description
BB	All displays, notifications, and online documentation of the identity management solution must be available in both English and Spanish.

8.3 Design approach

In this section we consider how security design objectives can be realized using Tivoli Identity Manager. Our goal is to produce a plan containing a phased set of implementation steps where the end result satisfies the functional requirements, and therefore also satisfies the original business requirements.

While business and functional requirements are the main parts of the security design objectives, we also must consider other non-functional requirements and constraints. These may include objectives that are necessary to meet general business requirements or practical constraints on constructing security sub-systems. Tivoli Identity Manager implementations often involve non-functional requirements relating to:

- ▶ High availability
- ▶ Backup and recovery
- ▶ Performance and capacity
- ▶ Change management
- ▶ Training
- ▶ Existing infrastructure
- ▶ Budget and staffing

Because we focus on the security architecture of identity management with Tivoli Identity Manager software in this book, we do not look in detail at all of these non-functional requirements.

To produce an implementation plan:

1. Prioritize the requirements.
2. Map the requirements to Identity Manager features.
3. Define the tasks involved in using those features to satisfy the requirements and estimate the effort required for each task.
4. Divide the tasks into phases.

Prioritizing the requirements is important because the priorities are one of the primary factors used to decide which implementation tasks will be done in which phase of the project. It is rare that an identity management solution can be created as a single deliverable satisfying every requirement. It is far more likely that it will be delivered in phases, and the highest priority requirements should be addressed in the earliest phases.

Assigning priorities to the requirements is often difficult because they are all important. You can more easily compare the priorities of requirements by asking questions that gauge the positive and negative impacts of the requirements:

- ▶ How much money will be saved when the requirement is met?
- ▶ Are there penalties if the requirement is not met?
- ▶ Is there a date by which the requirement must be met?
- ▶ Are there other requirements with dependencies on this one?
- ▶ If this requirement is not met, is the company any worse off than it is now?

After mapping the requirements to Identity Manager features and creating a list of implementation tasks, the requirement priorities and the effort of each task can be used to decide how to break up the project into phases. The goal of breaking the project into phases is to quickly deliver solutions to some high-priority requirements. This allows the company to begin seeing a return on their investment, while lower priority and more difficult tasks are still being executed.

8.4 Implementation approach

This section applies the design approach to TAA's specific requirements, as described in 8.3, "Design approach" on page 334.

Non-functional requirements

The non-functional design objectives are those that do not relate specifically to the functional requirements, but are items that should be addressed in the design. For TAA's project, these include:

- ▶ Re-use of the existing identity management infrastructure
- ▶ Standards to be used
- ▶ Maintainability and configuration management
- ▶ High availability and disaster recovery

Re-use of the existing infrastructure

The design must allow for the re-use of the existing identity management design, except where it conflicts with the new requirements. Furthermore, as highlighted in 7.2.4, "Secured e-business initiative architecture" on page 311, there is a future project to strengthen the security of the network architecture. Tivoli Identity Manager must therefore be deployed into the existing architecture in a way that allows the accommodation of network changes in the future with the least possible interruption to service.

Standards to be used

Where possible, the design must comply with standards in order to make subsequent implementation easier, more secure, and audit compliant.

Maintainability and configuration management

The design must allow for the maintainability of the system. This may involve deployment of some form of configuration management methodology or system management tool set.

High availability and disaster recovery

There are no additional requirements for high availability or disaster recovery over the first Tivoli Identity Manager project, so the design does not need to be concerned with these items. It is a good idea to be aware of the potential for future changes in directory choice and so on.

Requirement priorities

TAA has analyzed its business requirements and has made cost savings and timeliness of account creation and suspension its highest priorities.

Implementation tasks and efforts

The details of the implementation tasks are not described here. They are described in detail in the technical implementation chapters of this book, which are:

- ▶ Chapter 9, “Technical implementation: Phase I” on page 341
- ▶ Chapter 10, “Technical implementation: Phase II” on page 419
- ▶ Chapter 11, “Technical implementation: Phase III” on page 449
- ▶ Chapter 12, “Technical implementation: Phase IV” on page 511

Project phases

Based on the priorities of its business requirements and the levels of effort of the different implementation tasks, TAA has decided to split the project into four phases, as follows.

Phase 1: Installation and setup

The goal of this phase is to complete all of the work necessary to create an operational Identity Manager installation. At the completion of this phase, all of the Identity Manager components will be operational, and Identity Manager will have an accurate list of TAA employees and their accounts. This phase does not really address any of TAA's requirements, but it is a prerequisite to the work done in the following phases. Tasks in this phase include:

- ▶ Installation of Identity Manager and its required middleware components
- ▶ Definition of any custom person types
- ▶ Creation of an identity feed and validation of the feed data
- ▶ Installation of Identity Manager adapters

- ▶ Execution of reconciliations of each installed adapter to create a list of accounts and mapping to the owners
- ▶ Cleanup of any orphan accounts produced by the reconciliations (required for SOX compliance)
- ▶ Security hardening of the Identity Manager servers and components

Phase 2: Automatic account management

This goal of this phase is to generate a quick return on investment for TAA. It involves implementation tasks that address high-priority requirements, but that can also be completed very quickly. One of the factors governing how quickly a feature can be implemented and placed into production use is the number of people who must be trained to use the new feature. The implementation tasks for this phase were chosen because they are largely invisible to most employees. Very little training will be necessary beyond those people who are Identity Manager administrators. The features included in this phase are:

- ▶ Creation of common accounts (such as e-mail and Windows) for new employees
These accounts will be created automatically when a new person is created by the identity feed.
- ▶ Suspension of a person's accounts when the person is terminated
These accounts will be suspended automatically when the identity feed changes a person's status to an inactive value.
- ▶ Password synchronization using the Windows password change interceptor
When a user changes his Windows password, it will automatically change his other account passwords, too.

Phase 3: Delegated account management

The goal of this phase is to delegate account management activities to persons other than system administrators. This has been delayed to a later phase because the features implemented in this phase require many users to interact directly with Identity Manager. This requires the preparation of documentation and training for the delegated administrators. The implementation of these features also requires more extensive requirement gathering than the previous phases.

This phase includes many more features than any of the other phases. All of the features were included here because they meet the common criteria of delegating work from the system administrators to other users, but many of these features are independent of each other. There is no reason why this phase cannot be split into multiple sub-phases.

The features included in this phase are:

- ▶ Password self-reset using challenge/response questions

This feature is a good candidate to be moved into phase 2. It is quick and easy to implement, and it can result in a very large cost savings, but it does require giving all users Identity Manager accounts, teaching the users how to set their challenge answers, or teaching users how to use the feature to reset their passwords. TAA does not want to include any tasks in phase 2 that require training, so password self-reset was deferred to phase 3.
- ▶ Account management using the Identity Manager Web user interface

This enables Identity Manager to maintain a centralized audit trail regardless of whether the account management is being done by system administrators or by delegated administrators.
- ▶ Delegation
 - Account management

Users will be able to request the creation, modification, and deletion of accounts owned by persons whom they supervise. Account creation and modification require approval by a member of an administration team for the account's service.
 - Policy compliance

Users who transfer from one region to another may no longer be entitled to some of their accounts. A region's security and compliance team should be notified when this happens. They should be able to decide how long such a user will be allowed to retain his non-compliant account.
 - Group management

Where authority has been properly delegated, centralized system administrators and service owners will have the ability to create new groups, modify account membership of groups, and delete groups byway of the Identity Manager console interface.
- ▶ Change control for the Identity Manager configuration

The development of the phase 3 features cannot impact the correct functioning of the phase 2 deployment. The new feature development must be done in an isolated environment. New features must be migrated into the production environment as working units.

Phase 4: Role-based account management and separation of duty

The goal of this phase is to begin granting and removing access rights automatically based on a person's job role. This greatly reduces the chance for errors when granting and auditing access rights.

This can be a challenging phase to implement, as it is often not clear what roles exist within an organization. This phase is often left until the end of a project in order to allow sufficient time for the analysis of a business' job roles and the existing patterns of access rights.

The features that will be implemented by TAA in this phase are:

- ▶ Definition of roles, policies, and accesses for the following specific access rights:
 - User groups in corporate applications: All TAA's employees must have access to corporate applications only with user rights. Additionally, users should be able to request additional accesses based on their permissions and their entitlements.
 - Manager groups in corporate applications: All users with special access (manager access) must have access to corporate applications and must have manager group membership.
 - For the RACF infrastructure, TAA considers that only users in the central region access it.
 - Other special groups will be analyzed and assigned to the role that corresponds.
 - Identify role relationships that can be used to form role hierarchy structures.
 - Create separation of duty policies to manage roles that have been identified by TAA as having conflicting interests.
- ▶ Definition of roles and policies for the following generic system and application access:
 - Users from Tivoli Access Manager for e-business infrastructure
 - Users from Microsoft Active Directory infrastructure
 - Users from Lotus Notes infrastructure
 - Users from Linux infrastructure
- ▶ Requiring an annual review process where the managers recertify their employee's continuing need for their organizational role and group memberships. Additionally, accounts access on critical services must have the business need recertified monthly. This must be recertified by the employee first, followed by the employee's manager.

This concludes the design phase for the identity management project at TAA.

Archived



Technical implementation: Phase I

This chapter describes the tasks, considerations, and implementation details of the first phase of the Tivoli Identity Manager deployment at Tivoli Austin Airlines (TAA). These requirements are described in 8.2, “Functional requirements” on page 325.

Table 9-1 shows the mapping of the functional requirements (detailed in 8.2, “Functional requirements” on page 325) against implementation tasks detailed in 8.4, “Implementation approach” on page 335, that are relevant to this phase.

Table 9-1 Requirements for phase I

Functional requirement	Implementation tasks
Q. A central logging system is needed.	<ul style="list-style-type: none"> ▶ Installation of Tivoli Identity Manager and its required middleware components ▶ Installation of Tivoli Identity Manager adapters ▶ Completion of reconciliations of each installed adapter ▶ Generation of orphan accounts report ▶ Cleanup of any orphan accounts produced by the reconciliations
T. A regular feed of identity data from authoritative TAA sources into Identity Manager will be established.	<ul style="list-style-type: none"> ▶ Definition of any custom person types ▶ Creation of an identity feed and validation of the feed data
W. Stored sensitive data will be protected from unauthorized access.	<ul style="list-style-type: none"> ▶ Security hardening of the Tivoli Identity Manager servers and components
X. Transmitted sensitive data will be protected from unauthorized access.	<ul style="list-style-type: none"> ▶ Installation of Tivoli Identity Manager and its required middleware components ▶ Installation of Tivoli Identity Manager adapters ▶ Security hardening of the Tivoli Identity Manager servers and components
Y. The actions of Identity Manager users and administrators will be tracked in an audit trail.	<ul style="list-style-type: none"> ▶ Installation of Tivoli Identity Manager and its required middleware components
Z. Identity Manager administrators will not be able to manipulate the audit data or settings.	<ul style="list-style-type: none"> ▶ Installation of Tivoli Identity Manager and its required middleware components ▶ Security hardening of the Tivoli Identity Manager servers and components

Functional requirement	Implementation tasks
AA. Identity Manager components will be protected from access by unauthenticated or unauthorized users.	<ul style="list-style-type: none"> ▶ Installation of Tivoli Identity Manager and its required middleware components ▶ Security hardening of the Tivoli Identity Manager servers and components
BB. All displays, notifications, and online documentation of the identity management solution must be available in both English and Spanish.	<ul style="list-style-type: none"> ▶ Installation of Tivoli Identity Manager and its required middleware components ▶ Installation of Identity Manager language pack for Spanish

Each of the sections below can be traced back (either directly or indirectly) to the deployment requirements outlined in Table 9-1 on page 342 and detail the specifics required to achieve them. These are structured as follows:

- ▶ Information that is relevant during the design phase
- ▶ Design options and considerations
- ▶ How it was implemented at Tivoli Austin Airlines

9.1 Initial installation and configuration

The foundations of the solution must be built in order for further work to take place. This section focuses on the initial build of the software components required in an Tivoli Identity Manager deployment.

9.1.1 Requirements

The Tivoli Identity Manager middleware and application components must be installed and configured to allow for the design to be implemented.

9.1.2 Design considerations

Design considerations must be made for the software components and the physical architecture.

Software components

Tivoli Identity Manager supports the use of various Lightweight Directory Access Protocol (LDAP) repositories, relational databases, and application servers. Various reasons may be the determining factors in the selection of a particular software platform. These can range from corporate strategies and policies to feature/function evaluations. For example, the choice of IBM DB2 Enterprise Server as the relational database may be due to the fact that it is the corporate standard database server used by the organization.

Physical architecture

There are non-functional requirements to consider when designing the physical architecture of any solution. The initial phases of the project do not include non-functional requirements such as high availability, hence the initial build is to be kept simple with minimal redundancy built-in. For a more detailed discussion of non-functional requirements refer to Chapter 5, “Operational solution design” on page 223.

We follow best practices and make sure that operational and non-functional requirements are considered during the design of the initial architecture, even though not all of them will be implemented in the first phase. The following requirements are of special interest in this respect:

- ▶ Number of new servers required
- ▶ Operating system required on each new server
- ▶ Server sizing
- ▶ Software to be placed on new servers
- ▶ Software to be placed on existing servers
- ▶ Placement of new servers within the network

9.1.3 TAA's implementation

Based on the design considerations outlined, TAA made the following decisions.

Software

TAA made the decision to use the set of IBM products as the middleware components for the deployment, as their corporate strategy is to use IBM software solutions wherever possible. Hence, the full list of software components to be installed is:

- ▶ IBM Tivoli Directory Server
- ▶ IBM DB2 Universal Database™ Enterprise Edition
- ▶ IBM WebSphere Application Server
- ▶ Tivoli Identity Manager Base

- ▶ Tivoli Identity Manager Language Pack (for support of languages other than English)
- ▶ Tivoli Identity Manager Adapters (for each managed resource)
- ▶ IBM Tivoli Directory Integrator

Physical architecture

Due to the phased approach and the fact that the Tivoli Identity Manager infrastructure is to be built in a brand new environment, it is possible to use the production environment machines for development and to promote it to production status when development and testing is complete. The advantage of this approach is the reduced effort to create a production system. But there are also a few disadvantages. First, this approach can result in a production environment that may contain obsolete test data and configuration options left over from development and test activities. And second, this approach skips the process of migrating from development to production and therefore it lacks the chance to gain experience and test this process in the given environment.

Even though, due to time restrictions, the approach taken by TAA is to set up a development system first, a migration approach will be taken in later stages to build development and test environments based on the considerations outlined in Chapter 5, “Operational solution design” on page 223. TAA has provided a set of Active Directory, Tivoli Access Manager, Lotus Notes, Linux, and RACF test servers that mirror production to use during development and testing. The migration to production includes re-configuring any references to the test machines to use the relevant production equivalent.

Based on the requirements and Tivoli Identity Manager deployment best practices, the design team has determined that the production architecture be as shown in Figure 9-1. This does not contain an exhaustive list of all the machines in the TAA environment. It only lists the ones related to or affected by the Tivoli Identity Manager implementation.

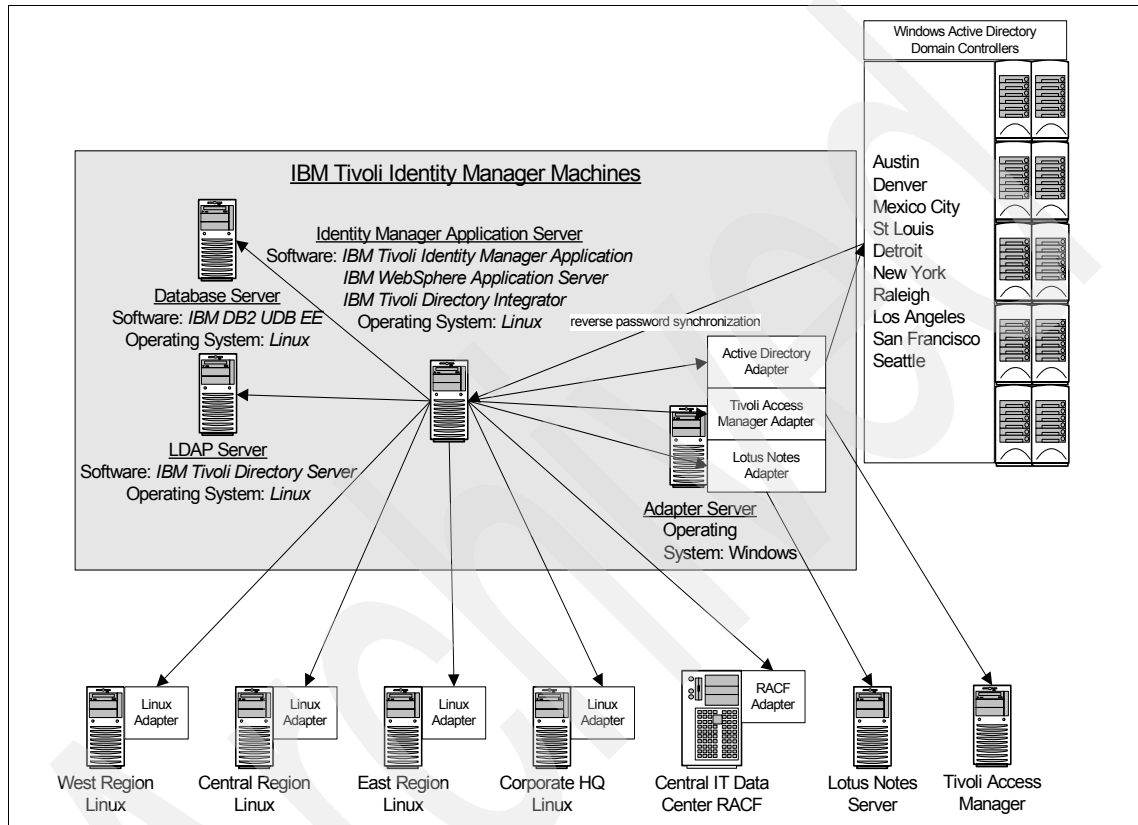


Figure 9-1 TAA Tivoli Identity Manager production physical architecture

The servers shown in Figure 9-1 on page 346 within the shaded box labelled *Tivoli Identity Manager Machines* are new machines to be procured for the deployment. All these new servers required for Tivoli Identity Manager are to be placed in the Central IT Data Center, as it is the most secure network zone currently available within the TAA network. The following list provides a brief description of each new server:

- ▶ Tivoli Identity Manager application server

The server on which the Tivoli Identity Manager application is to be deployed. The IBM Tivoli Directory Integrator component used for the identity feed is also deployed on this server. This Directory Integrator instance also runs the dispatcher service for Identity Manager adapters using the Java Remote Method Invocation (RMI) protocol, such as the Tivoli Access Manager Combo Adapter. Linux has been selected as the base operating system, as it is TAA's corporate directive to use Linux where possible.

- ▶ Database server

The Tivoli Identity Manager relational database is deployed on this server. Linux has been selected as the base operating system.

- ▶ LDAP server

The Tivoli Identity Manager LDAP server is deployed on this server. Linux has been selected as the base operating system.

- ▶ Adapter server

The Tivoli Identity Manager adapter components used to manage Lotus Notes and Microsoft Active Directory are installed on this server. These adapters do not need to be deployed on the managed resource, hence, as per best practice, they are deployed on a separate server to minimize the impact on the managed resource's environment. The decision has been made to use Windows as the base operating system, as the Microsoft Active Directory adapter cannot be run on any other operating system. This is due to the fact that it requires the necessary Windows native security permissions and access to the Active Directory APIs (ADSI) to be able to perform account management operations.

Machines not within the shaded box in Figure 9-1 on page 346 currently exist within the environment. That is, they are part of the current production infrastructure. These are:

- ▶ Windows Active Directory Domain Controllers

There are ten of these, as each CSC is a separate Windows domain. In phase II of the project an Tivoli Identity Manager reverse password synchronization component is installed on each domain controller to allow for Windows password changes initiated through Windows Active Directory to be managed by Tivoli Identity Manager.

- ▶ Tivoli Access Manager

This is the Tivoli Access Manager policy server that the Tivoli Identity Manager adapter communicates with to manage Tivoli Access Manager accounts. There do not need to be any Tivoli Identity Manager components installed on this server at this stage.

- ▶ Lotus Notes server

The Lotus Notes management server controlling the TAA e-mail system that the Tivoli Identity Manager adapter communicates with to manage Lotus Notes accounts. There do not need to be any Tivoli Identity Manager components installed on this server at this stage.

- ▶ Central IT Data Center RACF

This is the mainframe used by various TAA users. The Tivoli Identity Manager RACF adapter must be installed on this server. It cannot perform account management operations otherwise.

- ▶ Linux servers

This includes all four Linux servers:

- West region
- East region
- Central region
- Corporate HQ

Because the Tivoli Directory Integrator based Tivoli Identity Manager Linux adapter uses Secure Shell (ssh) to connect to the remote Linux servers, there is no need to install any Tivoli Identity Manager components on the Linux servers. Secure Shell is a standard feature that requires no additional software on UNIX platforms.

Server sizing

TAA has an agreement with IBM that allows for effectively priced solution deployments. This includes hardware and, as a result, their standard server build for a new server in their environment is an IBM System x® server with 500 GB

disk, 2 GB RAM, and a single or dual CPU processor. Based on the relatively low volume of users being managed by Tivoli Identity Manager (currently only managing internal TAA users) and consultation with Tivoli Identity Manager solution architects, this standard specification has been deemed to be acceptable for the deployment. As per the architecture diagram specified in Figure 9-1 on page 346, there is a need for four new machines. The decision has been made to have dual CPU processors on three out of the four machines. The adapter machine will have a single CPU, as the expected processing load on it is not expected to be as large as the other machines.

9.2 Secure the Tivoli Identity Manager application

Varying security measures should be taken to secure the system for the specific needs of the environment to minimize the security exposures present in the environment.

9.2.1 Requirements

TAA's functional requirements for the security of the Identity Manager application are shown in Table 8-9 on page 333.

9.2.2 Design considerations

In 5.5, "Security and integrity" on page 271, we describe the different areas of the Identity Manager installation that must be secured. This section discusses some of the options available.

Access to LDAP data

TAA is using an instance of the IBM Directory Server as Identity Manager's LDAP directory. IBM Directory Server's default behavior allows anonymous connections and read and search access to any connected user. TAA wants access to Identity Manager data to be controlled by Identity Manager ACIs when possible. Allowing casual browsing of the directory data might expose sensitive data. Even passwords that are stored encrypted would be more vulnerable to password cracking attempts.

One possible way to accomplish this goal would be to restrict access to the directory server's ports. This can be effective if the directory server can be isolated in a restricted network zone. But if people who should not have access to the Identity Manager data will have access to the directory server's network zone, or even to other data in the same directory server instance, then it will be necessary to configure access restrictions within the directory server.

Securing Identity Manager communications

Many of the connections between Identity Manager components can be secured using SSL. Figure 9-2 shows these connections.

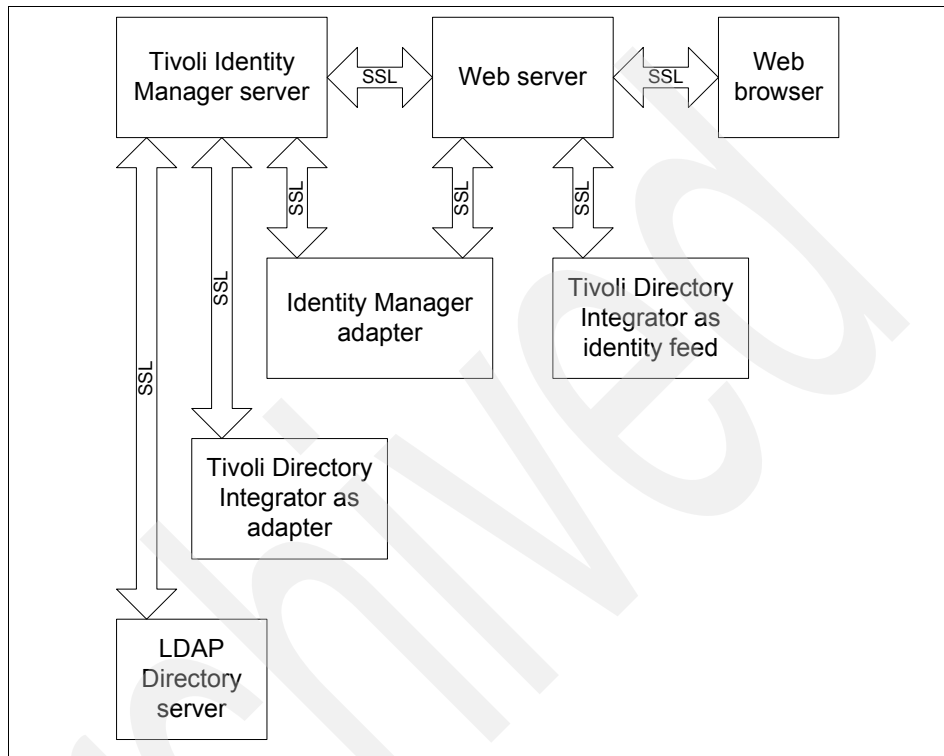


Figure 9-2 Identity Manager component connections with SSL support

The most critical of these connections from a security perspective are the connections from the Web browser to the Web server and on to the Identity Manager server, and the connections from the Identity Manager server to its adapters. These connections transmit account passwords, so the privacy of the data on these connections is very important. The remaining connections do not transmit clear text account passwords, but they do transmit credentials that are used for authenticating the Identity Manager components and may transmit sensitive user data.

It may not always be necessary to use SSL on a connection, even if the security of the data is critical. If two components are running on the same machine, or if they are connected by a secure subnet, it may be decided that SSL offers no additional benefit.

Note that there are two Identity Manager components that are not shown in Figure 9-2 on page 350. These are the Identity Manager database and optional Tivoli Identity Manager application clients. The Identity Manager server connects to its database using the JDBC protocol. Application clients may connect to the Identity Manager server using the IIOP protocol. Neither of these support SSL. This leaves limited options for securing these connections. These options include:

- ▶ Requiring that these components reside on the same machine
- ▶ Placing these components on machines connected by a physically secure network connection
- ▶ Using a tunneling protocol to create a secure private network between the components' machines

File permissions on the Tivoli Identity Manager server

The Identity Manager installation's data files should never be readable by any user except the file owner. Files in the data directory of the installation hold the key used to encrypt passwords stored by Identity Manager, and the user names and passwords used by Identity Manager to connect to its directory and database servers. Allowing an unauthorized user to read these files compromises the integrity of the Identity Manager system.

Passwords used by Tivoli Identity Manager

Identity Manager stores the passwords that it uses to connect to its directory and database servers in properties files in the data directory of the Identity Manager installation. The Identity Manager installation program gives you the option of storing these passwords encrypted or in clear text.

This may seem irrelevant if you have restricted read access to the Identity Manager properties files, but there is still a risk that an unauthorized person may see the passwords if they observe an administrator editing these files. Encrypting the passwords prevents an observer from learning these passwords.

Similarly, the key used to encrypt and decrypt passwords is entered during the installation process and stored in one of the properties files. This key is always stored in clear text. This value can be protected from observation by unauthorized users by using a long random value as the key. This makes it difficult for anyone who observes the key to remember it later.

Securing adapters

Identity Manager adapters face a number of security threats. The nature of these threats differs slightly depending on the type of adapter.

All adapters face the threat of false, or spoofed, requests. These are requests arriving over the network that falsely claim to have originated from the Identity Manager server. There are many steps that you can take to lessen this risk:

- ▶ Protect the adapter with a firewall that denies any connection requests on the adapter's port if the requests did not originate on the Identity Manager server.
- ▶ Change the adapter's default authentication ID and password to hard-to-guess values.
- ▶ Configure the adapter to use two-way SSL (SSL mutual authentication). One-way SSL uses a certificate to authenticate the adapter to the server, but relies on the authentication ID and password to authenticate the server to the adapter. Two-way SSL uses certificate-based authentication to authenticate the server to the adapter also.
- ▶ Protect the Identity Manager server's certificate. Someone with a copy of this certificate might be able to impersonate the server.

Adapters using the DAML protocol, such as the Lotus Notes and Microsoft Active Directory adapter, face an additional threat. These adapters are configured using a program named *agentCfg*. This is a network-enabled program. Any copy of the *agentCfg* program can connect to any adapter it can reach in the network. Once *agentCfg* has connected to an adapter and the correct agent authentication key is specified, it can use the adapter's self-test features to force the adapter to create and modify accounts. The steps that you can take to minimize this risk are:

- ▶ Protect the adapter with a firewall that denies any connection requests on the adapter's configuration port. This forces anyone running *agentCfg* to configure an adapter locally on the adapter host.

The first adapter to start on a host uses port 44970 as its configuration port. If multiple adapters are run on a single host, they each allocate a port. They start with 44970 and go up consecutively.

You can see which ports are being used as configuration ports on a host by running the following command on the host:

```
agentCfg -list
```

- ▶ Change the adapter's default configuration key.

The *agentCfg* program prompts for a key that is used to authenticate to the adapter. You can change this value by running the *agentCfg* program, connecting to an adapter, and selecting the option **D. Change Configuration Key**.

Some adapters must store credentials that they use to authenticate to the resource that they manage. When this is the case you must take steps to protect

these credentials. You can do this by restricting access to the adapter host and the file system where the adapter is installed.

9.2.3 TAA's implementation

The following sections describe how TAA has addressed the security of its Identity Manager installation.

Access to LDAP data

Identity Manager is configured to bind to its directory server as the `cn=root` directory user. This is the directory server's administrator user. TAA will remove access rights for all other users from the portion of the directory holding Identity Manager data.

Tivoli Directory Server's default permissions allow read and search access to the special `cn=anybody` user. Permissions granted to this user affect any user who binds to the directory server, regardless of whether the user has authenticated or used an anonymous bind.

TAA has created a new ACI removing this right at the root suffix containing the Identity Manager data. This ACI explicitly denies any access rights to `cn=anybody`. The effect is to deny any rights to all users except for `cn=root`. Figure 9-3 shows the access rights defined on TAA's `dc=com` object.

View access rights: o=otb,c=com

Subject DN:

Subject type:

Rights

Add child:

Delete entry:

Security class access rights

Security class	Read	Write	Search	Compare
normal	deny	deny	deny	deny
sensitive	deny	deny	deny	deny
critical	deny	deny	deny	deny
system	deny		deny	deny
restricted	deny	deny	deny	deny

Attribute access rights

Attribute	Read	Write	Search	Compare

Figure 9-3 Directory access rights for `cn=anybody`

Note: This example holds true for IBM Tivoli Directory Server Version 6.1. Other directory servers may have different default behaviors.

Securing Identity Manager communications

TAA is most concerned about the security of communications between users' Web browsers and the Identity Manager Web server, and communications between the Identity Manager application server and the Identity Manager adapters.

Users may be entering and retrieving passwords through the Identity Manager Web interface. These connections may be made through untrusted network zones, so it is critical that these connections always be secure. TAA ensures this by disabling HTTP connections to the Web server. Only HTTPS connections are allowed. The WebSphere Application Server hosting the Identity Manager application will also be configured to accept only HTTPS connections.

Disabling HTTP access to the Web server forces the use of HTTPS by the identity feed and by any adapters that are configured with event notification. Both of these communicate with Identity Manager through the Web server.

TAA will also use HTTPS connections to all of their Identity Manager adapters. These will be configured with two-way SSL authentication. This lets the Identity Manager server authenticate that it is talking to a known adapter, and lets the adapter verify that requests are coming from the real Identity Manager server.

TAA expects to eventually create a restricted network zone where the communications between the Identity Manager server, its directory server, and its database server will be protected. Until this is done they will protect the connection to the directory server with SSL. The connection to the database server will not be protected at this time.

TAA does not expect to deploy any application clients during this phase of its technical implementation. They may be used later to provide a simplified interface for user self-service. If TAA does deploy any application clients, they will require that they run on the same machine as the Identity Manager server.

File permissions on the Tivoli Identity Manager server

TAA has installed Identity Manager on a Linux server. It has protected the installation files from unauthorized viewing by removing any access rights for group and other users. It did this by executing the following command on the Identity Manager server:

```
chmod -R go-rwx /opt/IBM/itim
```

Before executing this command, the contents of the `/opt/IBM/itim/extensions` directory were copied to a public location. This allows the Identity Manager administrators to read the documentation and examples that are installed into this directory.

Securing adapters

TAA is securing their adapters by using two-way SSL authentication and by changing the configuration keys on each adapter. Each adapter's configuration key will be changed by the system administrator of the machine running the adapter.

9.3 Organization tree

An organization tree typically exists in any organization and at times it may exist in various forms. These can form the basis of any decisions made relating to the Tivoli Identity Manager organization tree design, but following this exactly does not usually result in a good design. For example, management and resource access are frequently handled geographically, so an organization tree based on the locations may be more useful.

9.3.1 Requirements

The organization tree is a key element of the entire Tivoli Identity Manager solution design. It provides containers for all objects within Tivoli Identity Manager, such as people, policies, services, and so on. Various aspects of Tivoli Identity Manager's design are controlled by the object placement within the tree.

9.3.2 Design considerations

A simple organization tree is much easier to manage, more adaptable to change, and lends itself to an easier implementation and maintenance. It is advisable to keep this tree as simple as possible. This typically involves a different type of approach to be taken from the one used to define the business' organization structure. In some cases it may make sense to map the business' organization tree into Tivoli Identity Manager, but more often than not, it is better not to. This is due to the fact that organization structures change frequently and modelling Tivoli Identity Manager based on it may cause unnecessary, frequent maintenance work to be performed with no added benefit. Also, identity management is not driven purely by business process models. It is a combination of business rules and security operational policies and procedures.

At a high level, the organization tree design is driven by:

- ▶ People's entitlements to accounts.
- ▶ What security policies for accounts are based on, for example, business unit or location.
- ▶ Administration of people, for example, what is the delegated administration model? Is it based on location administrators, centralized administrators, or business unit specific administrators?

The following Tivoli Identity Manager objects directly affect how these are enforced:

- ▶ Provisioning policies
- ▶ Password policies
- ▶ Identity policies
- ▶ Services
- ▶ Service selection policies
- ▶ Access control items

These objects can be placed within any container within the tree. It is the placement of these objects when combined with the business rules that determine where, how, and to whom a particular policy applies.

In Tivoli Identity Manager the organization tree does not have any impact on the usability of the graphical user interface in terms of a user and account administration tool. The user interface is designed to leverage a search function in order to get to any Identity Manager object easily. There is no need for drilling down through the organization tree to find objects like a person or an account. In other words, the organization tree does not serve any navigation purpose. This aspect, therefore, is of no interest when designing the organizational structure within Tivoli Identity Manager.

For more details on planning an organization tree see the IBM Tivoli Identity Manager Information Center Version 5.1, available at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_plan_orgtree.html

9.3.3 TAA's implementation

The main considerations for the design of the TAA organization tree are:

- ▶ How administration of people and their accounts is to be performed and by whom
- ▶ What types of accounts people have
- ▶ The physical locations of people's accounts

Taking a distilled view of TAA's security policies, it can be stated that:

- ▶ Administrators of one regional center can manage only users in that region.
- ▶ A manager of one division can do some administrative tasks on users in his division.
- ▶ Users in the CSC and regional centers log on to local Windows Active Directory domains.
- ▶ Linux accounts are only given to administrators located at the regional centers and all staff at corporate headquarters, according to Table 9-2 on page 360.
- ▶ Anyone can have Tivoli Access Manager accounts and Lotus Notes accounts.
- ▶ Only developers in the Central IT Data Center can have RACF accounts.

It is evident that TAA's security policy model is very location centric. As a result, the most simple approach is to use a location-based tree. The resulting TAA organization tree is shown in Figure 9-4.

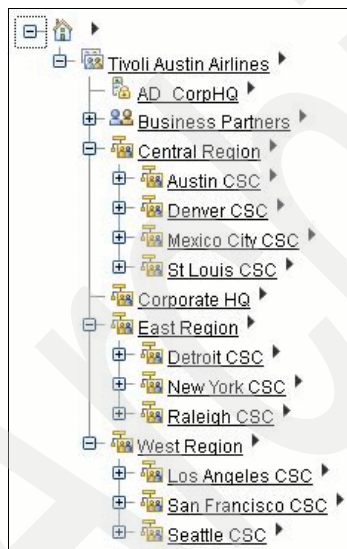


Figure 9-4 TAA organization tree

People in TAA are placed in the tree based on their relevant location. For example, a person located in the Denver CSC is placed in the *Denver CSC* organization unit. People who are not part of the CSC are placed in their region's organization unit. This includes regional administrators. For example, administrators for the central region are placed in the *central region* organization unit.

9.4 Services

Services define the managed resources containing the user accounts managed by Tivoli Identity Manager.

9.4.1 Requirements

A basic requirement underlying most of the functional requirements' outlines is that Tivoli Identity Manager must manage user accounts for the platforms detailed in TAA's requirements. To do this, Tivoli Identity Manager must know how to connect to the relevant managed resources. These are defined using services within Tivoli Identity Manager.

9.4.2 Design considerations

The main issues that must be considered here are:

- ▶ What are the managed resources and which accounts must be managed?

The first thing to determine is all of the platforms that must accounts managed. This should have already been considered when designing the physical architecture (refer to 9.1, "Initial installation and configuration" on page 343). There must be confirmation and refinement of the solution.

- ▶ What is the physical location of services?

Services are usually unique in an organization. For example, there is usually only a single RACF service, a single Tivoli Access Manager service, and so on. There may be systems where users are being defined according to location. A good example in this case is Windows Active Directory. Each location has its own domain and users must be provisioned to a particular domain depending on their locations.

- ▶ How are the services within Tivoli Identity Manager administrated?

Services can be centrally administered by an administrator or be delegated to sub-administrators to perform administration on a subset of services (the approach taken has potential implications on various aspects of the service design).

- ▶ What are the Tivoli Identity Manager actions for non-compliant accounts on services?

Tivoli Identity Manager can either mark the account, alert an administrator, suspend the account, or de-provision the account if it is non-compliant (that is, if it does not conform to the provisioning policies within the system).

These considerations affect the decisions about the location of the service definitions within the Tivoli Identity Manager organization tree. They should be placed in a part of the tree that avoids having to move them at a later stage. This is because a service location within the tree affects the scope of various other things such as provisioning policies, and such changes to a service can potentially have a multitude of behavioral implications. For example, a provisioning policy cannot apply to a service that is defined at a higher level in the organization tree. As a result, services must be at an equal or lower level in the organization tree than any provisioning policies that apply to them.

Some common approaches to service placement are:

- ▶ All services are placed at the organization tree root.

One of the advantages of this is that all services and provisioning policies can be found easily, as they are in a single location. This design is ideal for a centrally managed environment, although it is probably not a good idea if there are large amounts of services.

- ▶ Place services in admin domains.

If there is a distinct, logical separation of services into separate groups within an organization where they function completely independently of each other, this may be the approach to take. Delegating administration of services is straightforward, as there are less access controls to configure when compared to separating service access controls based on ACIs.

- ▶ Distribute services throughout the organization tree.

This approach is useful for ensuring that provisioning policies can only apply for the services that are relevant. Also, if services are separated into areas that conform to the structure of the organization tree, it may make sense to adopt this placement strategy. For example, if the organization tree is structured based on business units and service instances that are closely tied to the business units, then this may be an appropriate approach.

To further illustrate this example, consider two departments, finance and HR. Both departments may require access to a Windows domain. Standard users in both departments have exactly the same types of accounts. The difference is the domain in which they are created. Windows domain A services only the finance department, while Windows domain B services only the HR department. While it is possible to define both services at the organization tree root, having the services defined at the relevant parts of the organization tree can simplify the design. In the case of this example, only a single provisioning policy is required when used in conjunction with a service selection policy. The alternative is to define a provisioning policy for each service, which is not necessarily the most elegant approach when there are many services of the same type.

9.4.3 TAA's implementation

The full list of managed resources in the TAA environment to be managed by Tivoli Identity Manager is:

- ▶ RACF: For Central IT Data Center developers.
- ▶ Tivoli Access Manager: For all staff.
- ▶ Lotus Notes: For all staff.
- ▶ Linux: For administrators at each regional center and all staff at corporate HQ according to Table 9-2.
- ▶ Windows Active Directory: Same accounts for all entitled staff but different domains depending on physical location.
- ▶ Manual service: Employees in IT departments, as well as most of the back-office staff, require a PC for daily business. TAA plans to integrate the IT facilities department into the Identity Manager solution. The requirement is to coordinate the order and roll out process of IT equipment like PCs with the creation of new persons and accounts in managed resources. The TAA facility department does not work with an IT system that provides interfaces to external applications. That leaves no option but to build a custom adapter based on Tivoli Directory Integrator. Thus, the decision has been made to leverage the manual service feature of Tivoli Identity Manager in this case.

Additionally, some employees may also require a mobile phone as part of their employment. TAA plans to include mobile phone procurement requests in their Tivoli Identity Manager solution.

The general strategy adopted is to place services based on location and to have all non-compliant accounts on all services marked as non-compliant to prevent inadvertent de-provisioning of accounts. Table 9-2 lists all the services defined for TAA's environment. It makes references to the TAA organization tree. Refer to Figure 9-4 on page 357 for the full organization tree structure.

Table 9-2 TAA services

Service name	Service type	Location within organization tree	Reason
ITIM Service	Tivoli Identity Manager	Tivoli Austin Airlines	Every person has the potential to be provisioned with these accounts. This is required for the future phases that allow corporate-wide self-service features to be available. This is the default Tivoli Identity Manager service defined upon installation.

Service name	Service type	Location within organization tree	Reason
AlphaBeta Mobile Phone procurement	Manual service	Tivoli Austin Airlines	Some people require a mobile phone as part of their employment.
PC Procurement Service	Manual service	Tivoli Austin Airlines	There are people in every location that require PCs on which to work.
Tivoli Access Manager	Tivoli Access Manager	Tivoli Austin Airlines	Every person has the potential to be provisioned with these accounts. These accounts are not location specific.
Lotus Notes e-mail	Lotus Notes	Tivoli Austin Airlines	Every person has the potential to be provisioned with these accounts. These accounts are not location specific.
Central IT Data Center RACF	RACF	Tivoli Austin Airlines: Central region	Only developers in the central IT data center have the potential to be provisioned with RACF accounts. This means that RACF accounts are local to the central IT data center in Austin and there is no need to allow or manage RACF access for other locations.
Central Region Linux	Linux	Tivoli Austin Airlines: Central region	Administrative staff at the central regional center in Austin can have accounts on the Linux server located in Austin.
Corporate HQ Linux	Linux	Tivoli Austin Airlines: Corporate HQ	Staff at the corporate HQ site in Austin can have Linux accounts. This may be the same physical server as the one used by the administrative staff at the central regional center, but administration of the service within Tivoli Identity Manager is not necessarily governed by the same business rules. For example, if a future change in infrastructure requires a change of the service definition to use a separate Linux machine, it is the staff that has administrative rights on the corporate HQ environment that should have access to the definition within Tivoli Identity Manager to allow them to make the configuration change. It should not be the administrative staff in charge of the central regional center that have the ability to make configuration changes that affect corporate HQ staff.

Service name	Service type	Location within organization tree	Reason
East region Linux	Linux	Tivoli Austin Airlines: East region	Administrative staff at the east regional center in New York can have accounts on the Linux server located in New York.
West region Linux	Linux	Tivoli Austin Airlines: West region	Administrative staff at the east regional center in San Francisco can have accounts on the Linux server located in San Francisco.
Austin CSC Active Directory	Windows Active Directory	Tivoli Austin Airlines: Central region: Austin CSC	Staff located at the Austin CSC are the only people that who a requirement to be provisioned with accounts on the Austin Windows Active Directory domain.
Denver CSC Active Directory	Windows Active Directory	Tivoli Austin Airlines: Central region: Denver CSC	Staff located at the Denver CSC are the only people who have a requirement to be provisioned with accounts on the Denver Windows Active Directory domain.
Mexico City CSC Active Directory	Windows Active Directory	Tivoli Austin Airlines: Central region: Mexico City CSC	Staff located at the Mexico City CSC are the only people who have a requirement to be provisioned with accounts on the Mexico City Windows Active Directory domain.
St. Louis CSC Active Directory	Windows Active Directory	Tivoli Austin Airlines: Central region: St. Louis CSC	Staff located at the St. Louis CSC are the only people who have a requirement to be provisioned with accounts on the St. Louis Windows Active Directory domain.
Detroit CSC Active Directory	Windows Active Directory	Tivoli Austin Airlines: East region: Detroit CSC	Staff located at the Detroit CSC are the only people who have a requirement to be provisioned with accounts on the Detroit Windows Active Directory domain.
New York CSC Active Directory	Windows Active Directory	Tivoli Austin Airlines: East region: New York CSC	Staff located at the New York CSC are the only people who have a requirement to be provisioned with accounts on the New York Windows Active Directory domain.
Raleigh CSC Active Directory	Windows Active Directory	Tivoli Austin Airlines: East region: Raleigh CSC	Staff located at the Raleigh CSC are the only people who have a requirement to be provisioned with accounts on the Raleigh Windows Active Directory domain.
Los Angeles CSC Active Directory	Windows Active Directory	Tivoli Austin Airlines: West Region: Los Angeles CSC	Staff located at the Los Angeles CSC are the only people who have a requirement to be provisioned with accounts on the Los Angeles Windows Active Directory domain.

Service name	Service type	Location within organization tree	Reason
San Francisco CSC Active Directory	Windows Active Directory	Tivoli Austin Airlines: West Region: San Francisco CSC	Staff located at the San Francisco CSC are the only people who have a requirement to be provisioned with accounts on the San Francisco Windows Active Directory domain.
Seattle CSC Active Directory	Windows Active Directory	Tivoli Austin Airlines: West Region: Seattle CSC	Staff located at the Seattle CSC are the only people who have a requirement to be provisioned with accounts on the Seattle Windows Active Directory domain.

9.5 Provisioning policies

Provisioning policies govern the enforcement of people against the type of accounts on managed resources. Refer to “Provisioning policy” on page 118 and the IBM Tivoli Identity Manager Information Center Version 5.1, for more details, at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_admin_provisionpolicy.html

9.5.1 Requirements

Even though there are no specific requirements in this phase for business rules to be mapped to provisioning policies, there still must be basic definitions of policies to enable Tivoli Identity Manager to allow for people to own accounts. This is driven by the fact that the identity feed and reconciliation processes are to be performed in this phase. The identity feed results in people being loaded into Tivoli Identity Manager while the reconciliation processes for all services result in having accounts being owned by people. The lack of any provisioning policies causes Tivoli Identity Manager to deem that all accounts within the system are non-compliant. This can have consequences if not carefully planned for, for example, accounts being de-provisioned automatically by Tivoli Identity Manager if the service is configured to correct non-compliance.

At this stage of TAA’s implementation the Identity Manager system allows any type of account for every user. No compliance checking is required at this point in time. Accounts will not be corrected or de-provisioned automatically.

9.5.2 Design considerations

Given the requirements just mentioned above, no provisioning policies must be implemented to reflect business rules. The only design consideration to make is with regard to what policies are required to ensure that Tivoli Identity Manager allows people in the system to have accounts on the managed platforms.

9.5.3 TAA's implementation

The approach is to utilize the following two Tivoli Identity Managers features:

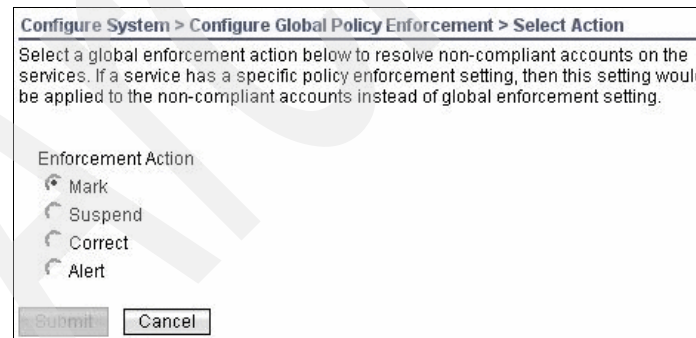
- ▶ Global policy enforcement
- ▶ Provisioning policy

For more details on each of these refer to the IBM Tivoli Identity Manager Information Center Version 5.1 at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_admin_provisionpolicy.html

Global policy enforcement

Global policy enforcement allows you to define a rule that is used as a default setting for *any* service within Tivoli Identity Manager. This global rule is obeyed by services unless there is a service-specific enforcement rule, which then takes precedence. In the case of TAA, the global policy enforcement is set to *mark on non-compliance*. This affects both non-compliant as well as disallowed accounts. It prevents accounts from being de-provisioned automatically. The global policy enforcement setting at TAA is shown in Figure 9-5.



The screenshot shows a web-based configuration window titled "Configure System > Configure Global Policy Enforcement > Select Action". The window contains the following text: "Select a global enforcement action below to resolve non-compliant accounts on the services. If a service has a specific policy enforcement setting, then this setting would be applied to the non-compliant accounts instead of global enforcement setting." Below this text is a section labeled "Enforcement Action" with four radio button options: "Mark", "Suspend", "Correct", and "Alert". The "Mark" option is selected. At the bottom of the window are two buttons: "Submit" and "Cancel".

Figure 9-5 Global policy enforcement

As a result of this all accounts being owned by people appear as disallowed, but will not be de-provisioned. Tivoli Identity Manager presents those accounts with a red status icon, as displayed in Figure 9-6.


Request... Change... Delete... Suspend... Restore... Assign to User... Refresh				
Se ^	State ^	User ID ^	Owner ^	Status ^
<input type="checkbox"/>		rhoffman	Rob Hoffman	Active
Page 1 of 1		Total: 1 Displayed: 1 Selected: 0		

Figure 9-6 Disallowed account owned by a person

Archived

Provisioning policy

Provisioning policies allow for the notion that anyone can have any account to be easily defined. This is achieved through the selection of the **All Services** target defined within the provisioning policy's entitlementment. In this case, a provisioning policy has been created at the root of the organization tree and is defined as shown in Figure 9-7, Figure 9-8 on page 367, and Figure 9-9 on page 367.

*General	
*Members	
*Entitlements	
	Manage Policies > Manage Provisioning Policies > General
	Specify information for the policy, the business unit to which the policy applies, and the scope of the policy within the organization. When you are done specifying information on each of the tabs, click Preview to review your changes, or click Save as Draft if you want to save your changes and finish this definition at a later time. Click Submit to save your changes now. Click Cancel to exit without saving your changes.
	*Policy name Allow All Accounts (Phase I)
	Caption <input type="text"/>
	Make policy available to services in <input checked="" type="radio"/> This business unit and its subunits <input type="radio"/> This business unit only
	Description <input type="text"/>
	Policy status <input checked="" type="radio"/> Enable <input type="radio"/> Disable
	*Priority (integer greater than 0) 1
	Keywords <input type="text"/>
	*Business unit Tivoli Austin Airlines <input type="button" value="Search..."/>

Figure 9-7 Allow all accounts provisioning policy General tab

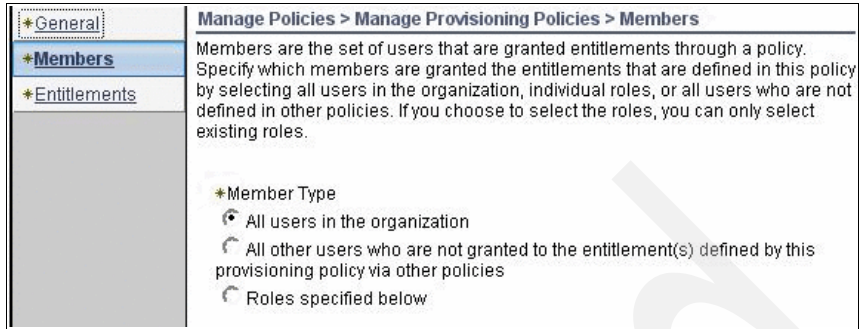


Figure 9-8 Allow all accounts provisioning policy Members tab

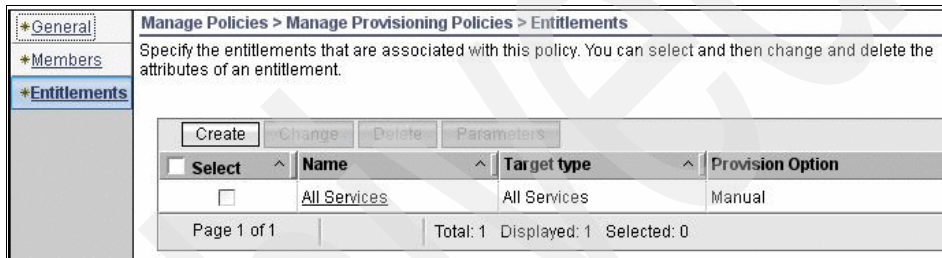


Figure 9-9 Allow all accounts provisioning policy Entitlements tab

Tivoli Identity Manager prevents provisioning policy entitlements of type *all services* to have any parameter. This is no limitation, but serves the purpose of allowing any account in any service to any person.

Figure 9-10 shows a sample account owned by a person as presented by Tivoli Identity Manager after applying this generic provisioning policy.

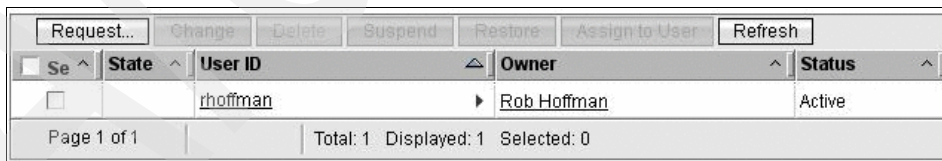


Figure 9-10 Compliant account owned by a person

9.6 Load users into Tivoli Identity Manager

This section discusses both the initial loading of persons into Identity Manager and the creation of a process for periodic updates to the person data.

9.6.1 Requirements

The creation of an automated feed of person data into Identity Manager satisfies a number of TAA's functional requirements. These requirements are shown in Table 9-3.

Table 9-3 *Functional requirements satisfied by identity feed*

Requirement	Description
C	Common values are entered automatically.
J	Automatically create common accounts when a person is employed.
K	Automatically add and remove accounts and access rights when a user changes job role.
T	A regular feed of identity data from authoritative TAA sources into Identity Manager are established.
U	An employee's accounts are disabled or removed when the identity feed shows that an employee has become inactive.

Only requirement T will be met during this phase of the implementation. Our goal in this phase is to maintain an accurate record of persons in Identity Manager. This is a prerequisite to completing the mapping of existing accounts to their owners, as discussed in 9.8, "Orphan account cleanup" on page 414. The identity feed that we create in this phase will be used in later phases to satisfy the remaining functional requirements.

9.6.2 Design considerations

The first thing to consider when designing an identity feed is the mechanism to be used for the feed. Identity Manager provides six possible choices:

- ▶ Comma-separated value (CSV) identity feed

This reads a comma-separated file to add users into Tivoli Identity Manager. The first record in the source file defines the attributes provided in each of the following records. All entries from the file are used in the feed process.

- ▶ Directory Services Markup Language (DSML) identity feed

DSML is an XML format that describes directory information. This was the preferred method for creating identity feeds in versions of Identity Manager prior to Version 4.4. DSML feeds must contain only valid attributes.
- ▶ AD organizational identity feed

This provides the capability for creating users based on user records from Windows Active Directory. Information from the AD organizationalPerson object class is mapped to the Tivoli Identity Manager Person profile schema. The attribute mapping file name option provides a way to customize the mapping of LDAP attributes to Tivoli Identity Manager attributes. This loads all user objects under a specified base.
- ▶ InetOrgPerson identity feed

This supports LDAP directory server using RFC2798 (inetOrgPerson LDAP object class). This loads all inetOrgPerson objects under a specified LDAP search base into the Tivoli Identity Manager Person profile schema. The attribute mapping file name option provides a way to customize the mapping of LDAP attributes to Tivoli Identity Manager attributes.
- ▶ Java APIs

You can use the Identity Manager APIs to manage person records. This offers the greatest flexibility of all of the methods, but is also the most complex and difficult way to create an identity feed.
- ▶ IBM Tivoli Directory Integrator identity feed

This is the most commonly used method for creating identity feeds. This method uses IBM Tivoli Directory Integrator to retrieve data from one or more sources. Use this data to add, modify, or delete person records in Identity Manager.

TAA makes the decision to implement the identity feed based on Tivoli Directory Integrator. The reason is that it allows you to validate data quality and add additional attributes (for example, persons manager) while reading the source data. Furthermore, Tivoli Directory Integrator offers the flexibility to integrate additional HR data sources without major changes to the existing concept at a later time.

The remainder of this discussion is specific to identity feeds created using Tivoli Directory Integrator. See “Identity feed management” in the IBM Tivoli Identity Manager Information Center Version 5.1, for more information about creating identity feeds.

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc/cpt/cpt_ic_idfeed.html

Once an identity feed method has been selected, the design will be influenced by the answers to the following questions:

- ▶ What data is available from the input sources?
- ▶ What data will be stored on Identity Manager person records?
- ▶ Will the feed process be initiated by Identity Manager or by an external process?
- ▶ What actions (for example, add, modify, delete, suspend, restore, transfer) will the feed perform on Identity Manager person records?
- ▶ Will the feed maintain relationships between person records and other Identity Manager objects?
- ▶ Are there different requirements for the initial load than there are for the ongoing feeds?

The following sections look at these questions in more detail.

Input data

There are some basic questions that must be answered about the input data for the identity feed:

- ▶ What attributes are available?
- ▶ Which attributes uniquely identify persons?
- ▶ Which attribute identifies a person's state? What states exist?
- ▶ Which attributes will always have values? Which might have multiple values?
- ▶ How are references to other person records, department numbers, and so on, represented in the data?
- ▶ How often is the data updated?

Sometimes the data source being proposed for the identity feed is not the authoritative source for some of its data. Changes to this data may take days or weeks to propagate to the identity feed's data source. This will not be acceptable if Identity Manager is expected to take quick action based on this data.

- ▶ In what format is the data available?

The data must be stored in a way that can be accessed using a Directory Integrator connector. There are standard connectors for database and directory servers and for data stored in files. Directory Integrator also has standard parsers available for many common files formats, such as XML, LDAP Data Interchange Format (LDIF), comma separated, and fixed column.

Even when the identity data is stored in an HR database, it is common for an organization's HR administrators to be reluctant to grant access to this database. An alternate way, like daily exports of HR data to a CSV file, can be considered. However, for the sake of reliability and easier operation of the synchronization interface, we strongly recommend direct access to the HR database or an appropriate API.

- ▶ How many data sources are there?

It is common for identity data to be stored in different locations. This may be a result of different HR systems in different parts of the organization, or different sources being maintained for employees, contractors, and customers. When there are multiple data sources, you must answer all of the preceding questions for each of the data sources.

Having multiple data sources is not necessarily a problem. Directory Integrator can read from two or more data sources as easily as it reads from a single source. Special care is to be taken if the multiple sources use different attributes to uniquely identify records, or if the sources contain similar data in different formats. For example, one data source may store full names in the format *first last*, while another data source stores full names in the format *last, first middle*. You must decide how to normalize the data if this is the case. Tivoli Directory Integrator provides multiple ways to handle this.

Output data

Many of the same questions that were asked about the input data must also be asked about the output data, such as:

- ▶ What attributes will be maintained on person records?

It is a common mistake to assume that every attribute available in the input data should be stored on the Identity Manager person records. This initially seems to simplify the decision-making process, but it actually forces you to make many more decisions than if you had minimized the amount of data stored on Identity Manager persons. Storing unnecessary attributes on Identity Manager persons has a number of drawbacks:

- It risks exposing sensitive data.
- It forces you to decide which person attributes will be visible in Identity Manager, and to whom they will be visible.
- You must make the same decisions about who can modify which attributes.
- You must design a custom person form that displays all of the attributes that will ever be visible.
- You must add your attributes to the schema of the Identity Manager LDAP server and create a custom object class to represent your person type.

- You must decide which of your attributes to index in the LDAP schema.
- The extra attributes will impact the size and performance of the LDAP server.

If storing too many attributes is bad, it is logical to ask how to decide which attributes should be included on the Identity Manager person objects. In general, the answer is to store only that data that will be used by Identity Manager. This is likely to include:

- Data that uniquely distinguishes each person.
 - Data that, while not necessarily unique, is used when describing a person.
 - Data that will likely be used by Identity Manager when provisioning accounts for a person.
 - Data that will be used to derive which type of process must be triggered to meet the use case requirements. As an example, the change of the persons status may trigger a simple *modify person* workflow, and it might also trigger a *suspend person with accounts* workflow.
 - Data that will likely be used by Identity Manager when assigning a person to roles or when placing the person in the organization tree.
- ▶ Which attributes are unique? Which are required? Which are multi-valued?
You must be sure that your answers to these questions match the answers for the corresponding input data.
 - ▶ How will missing values be handled? Are there default values?
 - ▶ Are there attributes that must be set on create, but not when modifying a person?

This will be the case if there are attributes for which the feed should provide a default value, but that may be modified later in Identity Manager. In this case the feed should not overwrite the changes that were made in Identity Manager.

How the feed is initiated

An identity feed using Tivoli Directory Integrator can be initiated by the Identity Manager server or by an external process. Both approaches use a Directory Integrator AssemblyLine to retrieve records from the input data source, and both use an *IBM Directory Integrator (IDI) data feed* Identity Manager service as the connection between Identity Manager and Directory Integrator.

Note: The Tivoli Identity Manager service type used in this section is still called *IDI data feed*. Although the current product name changed from IBM Directory Integrator (IDI) to IBM Tivoli Directory Integrator (TDI), the Tivoli Identity Manager service type name has not been changed.

When initiating the feed from the Identity Manager server, the feed is implemented as a reconciliation of an IDI data feed service. The reconciliation attempts to connect to an existing Directory Integrator process. When Directory Integrator gets a valid connection request it runs an AssemblyLine that retrieves the identity data from the input source. This data is returned to the Identity Manager reconcile operation. The data is then compared with existing Identity Manager person records. Existing records that differ from the data returned to the reconcile are modified. New person records are created when no existing record matched the input data. Since this approach involves Identity Manager asking that Directory Integrator send it data, this type of feed is sometimes called a *pull*.

When initiating the feed from an external process, that process must invoke a Directory Integrator AssemblyLine that retrieves the data from the input source and sends the data to Identity Manager. Directory Integrator sends the data to Identity Manager using a JNDI connection. This connection references an Identity Manager IDI data feed service. Since this approach involves an external process sending data to Identity Manager without being asked, this type of feed is sometimes called a *push*.

There are advantages and disadvantages to both approaches. These are summarized in Table 9-4.

Table 9-4 Advantages and disadvantages of different types of identity feeds

	Push feed	Pull feed
Advantages	<ul style="list-style-type: none"> ▶ Can delete people. ▶ Can take different actions on adds and modifies. ▶ Compare with existing record and calculation of delta is done in Directory Integrator. Identity Manager sees only new and modified data. ▶ Can be driven by events in the input data source. ▶ Can use an external scheduler. 	<ul style="list-style-type: none"> ▶ Can be scheduled or run on demand by any Identity Manager user with reconcile rights for the feed's service. ▶ Simpler to configure because there is no need for a JNDI connector in the Directory Integrator AssemblyLine. The person data is taken from the AssemblyLine work object without the need of an output connector at all.

	Push feed	Pull feed
Disadvantages	<ul style="list-style-type: none"> ▶ Scheduling a feed requires accesses outside of Identity Manager. ▶ More complex to configure because of the need for a JNDI connector. 	<ul style="list-style-type: none"> ▶ Cannot delete persons. ▶ Compare with existing records is done by Identity Manager server. ▶ Cannot be event driven. ▶ Requires that a Directory Integrator Listener process be waiting for connection requests.

What actions the feed will perform

Almost all identity feeds create new persons and modify existing persons. You must also consider whether your feed will ever need to delete, suspend, or restore persons.

If your feed must be able to delete persons from Identity Manager, this forces you to choose a *push* style of feed, where the Directory Integrator AssemblyLine uses a JNDI connector to send data to Identity Manager. A JNDI connector can issue delete requests to Identity Manager. A *pull* style of feed, initiated as a reconcile in Identity Manager, does not use a JNDI connector, and has no way to return delete requests to Identity Manager.

Suspend and restore operations can be done with either style of identity feed, but probably require that you customize the Identity Manager workflow that defines the modify operation for your person type. The reason for this lies in the command protocols used between Directory Integrator and Identity Manager. Depending on which style of feed you are using, the data is formatted as either JNDI or DSMLv2 operations. Neither of these command protocols has suspend or restore operations. A suspend or restore must be sent as a modify operation on the erPersonStatus attribute.

Simply modifying the erPersonStatus attribute does result in an Identity Manager person being labeled as active or suspended, but it does not cause any associated actions to be performed. So if you expect suspending a person to also suspend the person's accounts, know that this will not happen if all you do is modify the erPersonStatus attribute.

You can correctly simulate the suspend and restore operations by customizing the modify workflow for your person type so that it calls the suspend or restore operations when it detects that the `erPersonStatus` attribute is being modified. You can find instructions for performing this customization in the extensions directory of your Identity Manager installation. The instructions are located in `extensions/5.1/examples/workflow/erPersonStatusExample.html`. There is also an example of such a customization in 9.6.3, “TAA’s implementation” on page 380.

Relationships to other Identity Manager objects

An identity feed often must establish and maintain relationships between person records and other objects in Identity Manager. How you do this depends on the nature of the relationship, the way that the related object is specified in the input data, your object naming conventions in Identity Manager, and how important it is that your feed be optimized. The next two sections discuss the most commonly maintained relationships.

The parent relationship

The simplest relationship maintained by an identity feed is between persons and their parent containers in the organization tree. This is not done explicitly in the feed. Instead, a *placement rule* is defined on the IDI data feed service being used by the feed. The placement rule is a fragment of JavaScript code that is executed by the Identity Manager server once for each person record received from the feed. The placement rule uses the data received from the feed to calculate the person’s correct parent container. If an existing person is not in the container specified by the placement rule, then he is moved to the correct container.

The coding of the placement rule is usually a simple process. There are one or two attributes on a person that correspond with containers at specific levels in the organization tree. But there are some common situations that can complicate the placement of persons in the tree, such as:

- ▶ Some records in the feed are missing the placement attributes or have invalid values.

There are not many options for dealing with this situation. Identity Manager’s default behavior when a placement generates an error or returns a container specification that does not match an existing container is to place the person at the root of the organization tree. Sometimes this behavior is good enough. Identity Manager administrators should always be aware of person objects being created in the root of the tree, as this is often a sign of a broken placement rule or bad data in the identity feed.

Other options in this situation are to have the placement rule place persons with bad placement attributes in a special quarantine container or to make a best effort to get the person as close to the correct container as possible.

- ▶ The data values in the feed correspond to specific containers in the tree, but the data values in the feed are not equal to the names of the containers.

An example of this situation is where the organization tree containers are labeled with department names, but the feed data contains department numbers. There is a common misconception that placement rules must specify parent containers by name. In fact, a placement rule can specify containers using any attribute that is allowed on the container type. In the case of organizational unit containers, you may use any of the many attributes that are allowed on the standard LDAPv3 `organizationalUnit` object class. So for this example you can still label your container with department names, but also store the department numbers in the container's `businessCategory` attribute. Your placement rules can then specify the parent container as the one where the `businessCategory` attribute matches the department number in the feed data.

If you intend to use this technique you should be aware that Identity Manager's admin domain container type does not have any spare attributes in which you can store extra data. So you should plan your organization tree in such a way that you will not store person objects directly in admin domains.

See the topic "Determining the placement of the person" in the IBM Tivoli Identity Manager Information Center Version 5.1, for more information about creating placement rules.

The supervisor and role relationships

The supervisor relationship relates one person object to another person object. Identity Manager ACIs may grant a person the right to perform certain actions on the persons that they supervise or their accounts. The role relationship relates one person object to one or more Identity Manager organizational roles. These roles may grant the person the right to have certain accounts based upon the roles being members of one or more provisioning policies. The roles may also grant the person the ability to respond to approvals and other Identity Manager manual activities. These relationships are discussed together here because the manner in which they are managed by an identity feed and the problems you may encounter are similar.

Both the supervisor and role relationships are managed by setting special values on attributes in the identity feed. The supervisor relationship is set using the *manager* attribute (note that the actual LDAP attribute to use for the supervisor relationship can be specified via the person entity mapping entry for `erSupervisor`). The role relationship is set using the `erRole` attribute. Identity Manager knows that these attributes' values have special meanings when set by an identity feed. The attribute values are used to perform searches for the objects that satisfy the relationship with the person record being created or updated by

the feed. The steps taken by Identity Manager when it searches for supervisors and roles are similar, but there are some differences.

When an identity feed sets a value for a person's manager attribute, the Identity Manager server assumes that the value contains one of three types of data. It checks for these data types in the following order:

1. A distinguished name of a person object in the Identity Manager LDAP server

All person objects in the Identity Manager LDAP server use the attribute `erGlobalId` as their relative distinguished name (RDN[®]) attribute. So any manager attribute value beginning with "erGlobalId=" is assumed to be the distinguished name of the supervisor's person object.

2. One or more attribute names and values that can be used to construct an LDAP search filter

Any manager attribute value that does not meet the first criteria but does contain an equal sign (=) is assumed to be a list of attribute names and values. This list must follow the syntax of an LDAP distinguished name. The list will be split into its components, and each component will be combined into an LDAP filter using the ampersand operator (&). For example, if the manager attribute is set to the value "cn=Bob Smith,dept=100" the resulting search filter would be "(&(cn=Bob Smith)(dept=100))".

Once the search filter is created, the Identity Manager server first searches for a matching person starting at the container specified by the placement rules. The search descends into subtrees. If that search does not find a match, another search is done with the same filter, but starting at the root of the organization tree. The searches must return a single person to be considered successful. A person may have only one supervisor, so a search that returns multiple results is considered a failure.

3. A value that will match the name of the supervisor

Any manager value that did not meet the first two criteria is assumed to be the name of the intended supervisor. The supervisor is found using the same procedure described in step 2, but as though the manager attribute's value had started with "cn=".

Note: `cn` is the default naming attribute for the default person entity objectclass. For custom person classes, a different naming attribute can be specified, and this is the attribute name that will be assumed.

The processing of the role relationship is not quite as elaborate as that of the supervisor relationship. When an identity feed sets the `erRoles` attribute, the values are assumed to have one of two types of data. It checks for these data types in the following order:

1. A distinguished name of a role object in the Identity Manager LDAP server.

Any value containing an equals character (=) is assumed to be the distinguished name of a role.

2. A value that will match the name of a role.

The search for the role is done starting at the root of the organization tree. The role name must be unique. If the search returns multiple roles, the user will not receive membership in any of them.

Note that unlike the supervisor relationship, there is no way to specify an arbitrary search filter when setting a person's roles (this functionality was not provided, as the `erRole` objectclass only has one other attribute, `description`). Also note that the `erRoles` attribute may have multiple values. The search process described above is used for each of the attribute's values. And finally, keep in mind that setting a person's roles in an identity feed overwrites any role memberships that they already have in Identity Manager. Your feed would need to retrieve the person's current roles from the Identity Manager LDAP server and merge those with any new roles if you want to add roles to a person without replacing the originals.

Note: Roles could be organized into role hierarchies. This should be considered during the planning phase. For further discussion, refer to 12.1, "Preparing for role-based access control" on page 512.

Relationships and unnecessary modify requests

One of the advantages to using an identity feed that pushes data to Identity Manager using Directory Integrator's JNDI connector is that this connector attempts to avoid sending modify requests if the person data from the input source matches the person record in Identity Manager. This reduces the load on the Identity Manager server when only a small percentage of the person records have updates.

The JNDI connector does this by issuing lookup requests to Identity Manager's JNDI service for each record in the input data. If no record is found by the lookup, the JNDI connector issues an add request to Identity Manager, and a new person record is created. If the lookup returned data from an existing person record, this data is compared with the values in the identity feed's input data. The record is skipped if all of the values are equal. Otherwise, a modify request is issued to Identity Manager for just the attributes needing updates.

This technique breaks down if the feed is setting supervisor or role relationships using any values other than distinguished names. A person's supervisor and role memberships are always stored as the distinguished names of the related objects, regardless of whether they were originally set by passing the supervisor's and roles' names. So when the same feed is run again, the manager and erRoles attributes in the input data are set to the same values as before. But the lookup done by the JNDI connector returns the distinguished names of the supervisor and roles. The distinguished names are not equal to the object's display names that are in the input data, so the connector believes that it must modify this person.

When this modify request arrives in the Identity Manager server, it follows its procedure for finding the person's supervisor and roles. It finds the same objects that were found during the last feed, so nothing changes.

The processing of the manager and erRoles attributes is more expensive than that for any other attributes that can be set through an identity feed. So having every record in the feed detect a change on these two attributes has the potential to greatly slow the feed and any other processing being done by the Identity Manager server at that time.

There are two options available to you for handling this issue. The first is to ignore the issue. If you do nothing the results of your feeds will be correct. It just takes longer to get the results. This is not different than if you were using the style of feed that uses a reconcile of the IDI data feed service to pull data from Directory Integrator. In that style of feed all of the input data is sent to Identity Manager without checking for persons who are unchanged.

The other option is to always set the manager and erRoles attributes to the full distinguished names of the supervisor and role objects. This ensures that if the person is unchanged the values in the input data will match the values returned by the lookup of the person's current values in the Identity Manager LDAP server.

If you take this second option, you must replace the search logic being used by Identity Manager to find the related objects with equivalent logic in your Directory Integrator AssemblyLine. You will not be able to do your search with a JNDI connector like the one used to send the person updates to Identity Manager. The Identity Manager JNDI service's lookup operations can be used only for person searches, so it cannot be used for finding a person's current roles. Although you could use the Identity Manager JNDI service to search for a person's supervisor, the JNDI service does not return the found object's real distinguished name. These limitations in the Identity Manager JNDI service force you to connect directly to the Identity Manager LDAP server to search for a person's supervisor and roles. This connection to the Identity Manager LDAP server can be made using Directory Integrator's LDAP connector type. A sample implementation of this technique is shown in "The GetManager connector" on page 396.

Considerations for initial loads

The initial load of existing persons often must behave differently from a day-by-day feed of new and modified persons. For example, the requirements for the ongoing feed may call for the feed to assign a unique ID and e-mail address to each new person added to Identity Manager. Doing this during the initial load would be a mistake because what appears to be new persons are actually existing persons who already have IDs and e-mail addresses.

During the initial load we want to replace the calculation of new unique values for this ID and e-mail address with a lookup of the person's existing values. How this data is collected and made available to the initial load will be different for each deployment of Identity Manager.

Another consideration during initial loads is the handling of dependencies between the person records. If you are maintaining references to persons' supervisors, you may load person records that refer to a supervisor before that supervisor's record has been loaded. Sorting the input records to eliminate forward references is probably a difficult and time-consuming task. The easiest way to address this problem is to perform the load in two stages. The first stage loads all person data except for the supervisor relationship, while the second stage loads only the supervisor relationships.

You may want to take a similar approach with the ongoing feed. It is possible that two new persons will appear in the same feed, and that one will be the other's supervisor. In reality this should be a rare occurrence. Even when it does happen, the correct supervisor relationship would be created by the next feed. The low probability of this happening, and the low impact if it does happen, means that this issue is often not addressed in ongoing feeds.

9.6.3 TAA's implementation

This section describes the specific identity feeds created to meet TAA's functional requirements. The following sections describe:

- ▶ The input data available for the feed
- ▶ How this data is mapped to Identity Manager person attributes
- ▶ How Directory Integrator is configured to implement the feed
- ▶ How this feed is modified to perform the initial load of existing persons

The identity feed's input data

TAA's human resources department intends to perform a nightly export of data from its database. This export file contains CSVs and a record for every employee in the database. The records in the CSV file contain the fields listed in Table 9-5 on page 381.

Table 9-5 Fields exported from the human resources database

Field	Description
employeeid	A numeric value guaranteed to be unique for each person.
givenname	First name.
sn	Last name.
initials	Middle initials.
employeetype	An integer value indicating the job type of the employee.
l	A unique key that identifies the location of the employee.
locale	The ISO 639 language code for the person's preferred language. This is either en for English or es for Spanish.
status	Employee status. Valid values are: A Active L Leave of absence I Inactive
supervisorId	Employee ID of the person's supervisor. Blank if none.
startdate	Date of start of contract in format YYYYMMDD.
enddate	Date of end of contract in format YYYYMMDD.

Each export file begins with a line containing the names of the fields. This is followed by the exported data with one line per person. Example 9-1 shows a small sample of an export file.

Example 9-1 Sample identity feed input data

```
employeeId,givenname,sn,initials,employeetype,l,locale,status,supervisorId,start
tdate,enddate
4711,Robert,Hoffman,B,4000,RE,en,A,0815,20080406,20091231
```

The IDI data feed service

The identity feed requires the use of an Identity Manager service with a type of IDI data feed. TAA has created such a service using the parameters shown in Figure 9-11.

Manage Services > Change a Service > Service Information

To change the service information, make the required changes, and then click OK. To test the connection to the service, click Test Connection.

*Service name
HrFeed

Description
Daily feed of HR data

*URL
http://not_used_here:8800

User ID
agent

Password

*Naming context
dc=HrFeed

Use workflow

*Name attribute
employeeNumber

Placement rule
return "|=" + entry.I[0];

Figure 9-11 Identity feed service details

Note that the URL field is not set to a valid URL. This value is not used when feeds are initiated from an external process. It is only used when a feed is initiated by a reconcile of the service. When that is done, the URL must specify the host and port on which Directory Integrator is listening for connection requests. TAA will always initiate their feeds from external processes, so the mandatory URL field can be set to any placeholder value.

The Use Workflow field on the service form controls whether the identity feed creates and modifies persons by direct manipulation of the Identity Manager directory or by creating person add and modify requests. TAA is running its identity feeds without workflow during this phase of the implementation. They do not want the feed to cause provisioning policy enforcement or suspension of

accounts until they have verified that the feed is executing correctly, and that account ownerships have been assigned accurately.

The placement rule places persons in the Identity Manager organization tree based on the value of their l (location). The location field in the feed's input data contains the unique key that identifies the location at which the person works. At this point it pays off that TAA has implemented a relatively flat organization tree. The placement rule that is required in this case is straightforward and shown in Example 9-2.

Note: This script assumes that every feed entry has an l (location) attribute and will fail if any entry is missing this attribute.

Example 9-2 Placement rule for TAA's identity feed service

```
{ return "l=" + entry.l[0]; }
```

Mapping input fields to person attributes

Most of the fields in the input file simply map to Identity Manager person attributes of the same or a similar name. A few require special handling. These attributes are described in the following sections.

Preferred language

Tivoli Identity Manager allows a user to work in her preferred language. When a user selects a language, all Identity Manager windows, e-mail notifications, and online help appear in that language.

Users have a number of options for selecting their preferred language:

- ▶ Click the **Select another language** link on the Identity Manager login page.
This would be bothersome to do every time that you log in to Identity Manager. It is not even an option if Identity Manager is configured for single sign-on and the login page is bypassed.
- ▶ Set a language preference in the browser.
This delivers the language-appropriate logon page, but it does not affect the language used when e-mail notifications are sent to the user.
- ▶ Set a language preference in the erLocale attribute of their Identity Manager person object.

This controls the language used whenever an e-mail notification is sent to the person. It also controls the language used in Identity Manager sessions once the user has logged on.

We use the locale field in the feed's input data to set the erLocale attributes of newly created persons. But the feed does not set this attribute when a person is modified. This allows users to change their preferred language in Identity Manager without the identity feed overwriting their new selections.

TAA employee status

A TAA employee may be in one of three states, as shown in Table 9-5 on page 381. Identity Manager has only two person states:

- ▶ Active
- ▶ Inactive

The identity feed must somehow map the TAA employee states into the two Identity Manager person states.

TAA has made the obvious decision that the active and inactive employee states will map to the active and inactive person states. They have also decided that employees who are on a leave of absence will remain active in Identity Manager.

This mapping could be made directly from the values of the status field in the input data to corresponding values of the erPersonStatus attribute on the Identity Manager person objects. But TAA has decided that they may want to take different provisioning actions for active employees and those on leave. For this reason, it has been decided that the TAA employee status must be stored on the Identity Manager person objects.

TAA employee start of contract and end of contract dates

The person data exported from the TAA HR database contains two valuable attributes that indicate the start and end dates of an employees contract. Both dates are used to trigger appropriate life cycle operations within the Identity Manager solution. Therefore they must be stored on the Tivoli Identity Manager persons object.

Creating a custom Tivoli Identity Manager person object

Identity Manager's person objects are based on the LDAPv3 standard's inetOrgPerson object class. This object class does not provide attributes appropriate for storing the following attributes that are specific to the TAA environment:

- ▶ Employee status
- ▶ Start of contract
- ▶ End of contract

To meet the requirements TAA must define its own custom person type with three new attributes that can hold this data. To accomplish this:

1. Create a new attribute schema in the Identity Manager LDAP server.

The new attribute names are `taaEmpIStatus`, `taaStartOfContract`, and `taaEndOfContract`. All of these are defined as single value attributes that store case-insensitive strings. They are indexed for optimized searching at later stages.

2. Create a new object class schema in the LDAP server.

The new object class' name is `taaEmployee`. It uses `inetOrgPerson` as its superior class. It adds `aaEmpIStatus`, `taaStartOfContract`, and `taaEndOfContract` as allowed attributes.

Archived

3. Create a new person entity type in Identity Manager.

The new entity's name is Employee. It uses taaEmployee as its underlying LDAP object class. The default erSupervisor mapping was changed from the erSupervisor attribute to the manager attribute. This is necessary for the correct operation of ACIs that grant privileges to supervisors. See the mappings of the person entity for the correct mappings. Figure 9-12 and Figure 9-13 display the new employee person object as defined in Tivoli Identity Manager.

Manage Entities > Change Entity > Entity Detail Information

Type the entity schema information. You must click Search to specify the super class. When you are done specifying information on each of the tabs, click OK.

*Entity name: Employee

*LDAP class: taaEmployee

*Name attribute: cn

Default Search attributes:

- audio
- businesscategory
- carlicense
- departmentnumber
- description
- destinationindicator
- displayname
- employeetype
- facsimiletelephonenumber
- givenname

Browse name attributes...:

- cn
- employeenumber
- uid

Figure 9-12 TAA employee definition

Select	Identity Manager attribute	Custom LDAP attribute
<input type="checkbox"/>	cn	cn
<input type="checkbox"/>	eraliases	eraliases
<input type="checkbox"/>	ercustomdisplay	sn
<input type="checkbox"/>	erlastoperation	erlastoperation
<input type="checkbox"/>	erlocale	erlocale
<input type="checkbox"/>	erroles	erroles
<input type="checkbox"/>	ersharedsecret	ersharedsecret
<input type="checkbox"/>	ersupervisor	manager
<input type="checkbox"/>	mail	mail

Page 1 of 1 | Total: 9 Displayed: 9 Selected: 0

Figure 9-13 TAA employee attribute mapping

4. Define a form to be used when displaying employee objects in Identity Manager.

The form design must be done in such a way that the form labels are displayed in the correct language for the current user. Each form field and tab have a label property. Figure 9-14 shows the properties for the `taaEmpIStatus` field, as shown in the Identity Manager form designer applet. Note the value of the Label field. If the label value begins with any character other than a dollar sign (\$), the value is used as the label on the form. The dollar sign at the start of the label value indicates that the remainder of the value is a property name. This property name will be used to do a lookup of the actual label text. The lookup will be done in a properties file in the data directory of the Identity Manager installation.

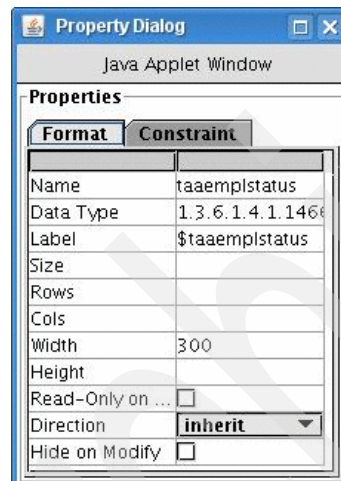


Figure 9-14 Editing a field's properties in the Identity Manager form designer

Which properties file is used for the lookup depends on whether any Identity Manager language packs have been installed, and what language is selected by the current user. With no language packs installed the lookup is done in the `CustomLabels.properties` file. With language packs installed the lookup is done in a language-specific properties file based on the current user's selected language. The language-specific files have two-letter ISO 639 language codes appended to the file names, with optional two-letter ISO 3166 country codes appended to that. Example properties files for specific languages and locations are:

English	<code>CustomLabels_en.properties</code>
Spanish	<code>CustomLabels_es.properties</code>
German	<code>CustomLabels_de.properties</code>

Each label property added to a form should be defined in all of your labels properties files. A definition consists of the property name followed by an equals sign (=) followed by the text that you want to appear on the form.

The changes that TAA has made to its CustomLabels_en.properties file are:

```
# Tivoli Austin Airlines (taa) labels  
taaemplstatus=Employee Status  
taaendofcontract=End of Contract  
taastartofcontract=Start of Contract
```

The changes that TAA made to its CustomLabels_es.properties file are:

```
# Tivoli Austin Airlines (taa) labels  
taaemplstatus=Estado de empleado  
taaendofcontract=Fin de contrato  
taastartofcontract=Inicio de contrato
```

Identity Manager person status

“TAA employee status” on page 384 explained why TAA wanted to store its employees’ states and why this led it to create a custom person entity type in Identity Manager. It also wants to map its employee status values to the standard Identity Manager active and inactive person states.

Identity Manager stores a person object’s state in its erPersonStatus attribute. This attribute will be missing, or have a value of 0, for active persons. It will have a value of 1 for inactive persons.

The erPersonStatus attribute can be set by the identity feed, but doing this can cause two potential issues that must be addressed:

1. The Identity Manager JNDI service does not return a person object’s erPersonStatus attribute when a lookup request is done. So if the feed attempts to set this attribute on a person, the JNDI connector does not see any existing value for the attribute and believes that the person’s data has changed. This can result in many unnecessary modify requests being sent to Identity Manager.
2. The erPersonStatus attribute is an internal attribute that reflects results of Tivoli Identity Manager person entity processes such as add, suspend, and delete. Changing the attribute by the identity feed without triggering the appropriate workflow does not make sense in most cases. In fact, it could lead to inconsistency between what is shown on the person object as opposed to the current status of the persons accounts.

Since the concept at TAA is to store the person's status as provided by the HR department in a custom attribute `taaEmplStatus` rather than the standard `erPersonStatus`, it is easy to address both of those issues. The approach here is to have the identity feed set the `erPersonStatus` attribute only for new employees, that is, on *add person* operations only. On any subsequent modify operation the identity feed ignores the `erPersonStatus` attribute, thus avoiding unnecessary modify requests on behalf of it. The `erPersonStatus` attribute is updated through Identity Manager life cycle operations only. This makes sure that it contains valid data at each point in time.

This is shown in detail in “The WritePerson connector” on page 396.

Finding the person's supervisor

The input data references a person's supervisor by the supervisor's employee number. We use this value to look up the supervisor's distinguished name in the Identity Manager LDAP server and pass the distinguished name to Identity Manager as the person's manager value.

Generating a unique ID and e-mail address

TAA assigns a unique ID to each person. This ID is used as the person's account name on all platforms. The ID is also used as the local part of the person's e-mail address.

The ID is generated by using the person's first initial and last name. If this value is already in use, numbers starting with two and incrementing from there are appended to the end of the ID until a unique value is found. A person's e-mail address is her ID plus `@taair.com`.

The identity feed could generate both of these values, but this would not assign values for any Identity Manager persons who are not created through the identity feed. If administrators manually create person objects for contractors or other users who are not in the human resources database, then these people also need unique IDs.

To ensure that manually entered persons also get a unique ID, TAA has decided to place the generation of the IDs in the *add* workflow for its custom employee type. In this case it is mandatory to activate the *use workflow* option on the HR feed service definition to make sure that unique IDs and mail addresses will be allocated for persons created by the identity feed. The updated HR feed service configuration is shown in Figure 9-15.

Change Service

Manage Services > Change a Service > Service Information

To change the service information, make the required changes, and then click Test Connection to the service, click Test Connection.

*Service name
HrFeed

Description
Daily feed of HR data

*URL
http://not_used_here:8800

User ID
agent

Password

*Naming context
dc=HrFeed

Use workflow

*Name attribute
employeeNumber

Placement rule
return "l=" + entry.l[0];

OK Cancel Test Connection

Figure 9-15 HR feed service with use workflow option

The original *add* workflow is shown in Figure 9-16 on page 391, while the customized workflow is shown in Figure 9-17 on page 391.

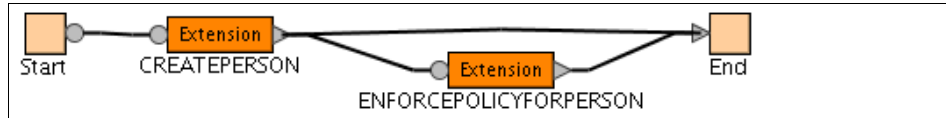


Figure 9-16 Original add workflow for employee entity

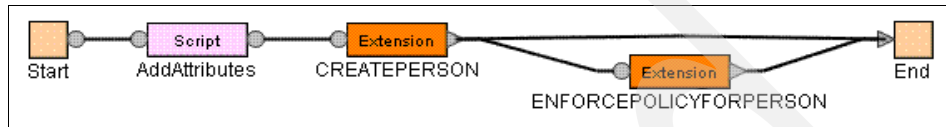


Figure 9-17 New add workflow for employee entity

The script node that is added to the workflow contains the code shown in Example 9-3. This JavaScript first looks for the presence of a uid attribute in the add request. If none is present, the script calculates a unique value and adds that value to the request. It then does the same for the mail attribute. If this attribute is not present in the request, the script adds it with a value based on the uid.

Example 9-3 Workflow script to set uid and mail attributes

```

// EmployeeAdd_Start
//
var p = person.get();
var str = "";
var uid = "";
var smtpDomain = "@taair.com";
// get current date
var curDateTime = Enrole.toGeneralizedTime(new Date());
var defDateEnd = "204912311200Z";

function isNull(s) {
  if ( (s != null) && (s.length > 0) && (s[0].trim() != '') ) return false;
  else return true;
}

//=====
// set default values
//=====

str = p.getProperty("taaStartOfContract");
if ( isNull(str) ) {
  // missing attribute
  p.setProperty("taaStartOfContract", curDateTime );
}

str = p.getProperty("taaEndOfContract");
  
```

```

if ( isNull(str) ) {
    // missing attribute
    p.setProperty("taaEndOfContract", defDateEnd );
} else {
    if ( str[0] < curDateTime ) {
        p.setProperty("taaEndOfContract", curDateTime );
    }
}

//=====
// make sure that person has a unique id
//=====

// check if uid has been provided by HR Feed or GUI
str = p.getProperty('uid');
if ( !isNull(str) ) {
    // use uid as provided
    uid = str[0].toLowerCase();
} else {
    // uid is missing, so let's generate a new one
    // get first char of givenname if available
    str = p.getProperty('givenname');
    if ( !isNull(str) ) {
        uid = str[0].substring(0,1).toLowerCase();
    }

    // add surname to first char of givenname
    uid += p.getProperty('sn')[0].toLowerCase(); //mandatory attribute
}

// Search for anyone who already has this uid. Keep looping until a
// unique value is found.
var uidBase = uid;
var count = 1;
var idInUse = true;
var personSearch = new PersonSearch();

while ( idInUse ) {
    var results = personSearch.searchByFilter('Employee', '(uid=' + uid + ')',
2);
    if (( results != null ) && ( results.length > 0)) {
        // found existing entry that has this uid already
        // continue searching
        uid = uidBase + count++;
    } else {
        // found free uid
        // stop searching
        idInUse = false;
    }
}

```

```

}
p.setProperty('uid', uid);

//=====
// use uid as localpart of that persons mail address
//=====

// check if mail address has been provided by HR Feed or GUI
str = p.getProperty("mail");
if ( isNull(str) ) {
    // use uid as localpart of mail address
    p.setProperty('mail', uid + smtpDomain);
}

// finally update person object in WF context
person.set(p);

```

Note that this method of generating IDs is not guaranteed to generate unique values. It is possible that two threads could both be testing the same value for uniqueness in parallel, and that both would find the value unique before both assigning the same value to different persons. TAA considers this situation to be so unlikely that it does not address the issue at this point in time.

Implementing the identity feed using Directory Integrator

This section makes extensive references to the features and capabilities of IBM Tivoli Directory Integrator.

For more information about the use of this product, see the following documents:

- ▶ *IBM Tivoli Directory Integrator 7.0: Getting Started Guide*, GI11-8185
- ▶ *IBM Tivoli Directory Integrator 7.0: Users Guide*, SC23-6561
- ▶ *IBM Tivoli Directory Integrator 7.0: Reference Guide*, SC23-6562

Note: For reference purposes we include the Directory Integrator AssemblyLine used in the TAA identity feed approach. For more details on how to obtain the configuration file refer to Appendix D, “Additional material” on page 655.

TAA's ongoing identity feed is implemented as a single Directory Integrator AssemblyLine. This AssemblyLine consists of the three connectors and one IF branch, shown on the data flow tab of the Directory Integrator configuration editor window in Figure 9-18.

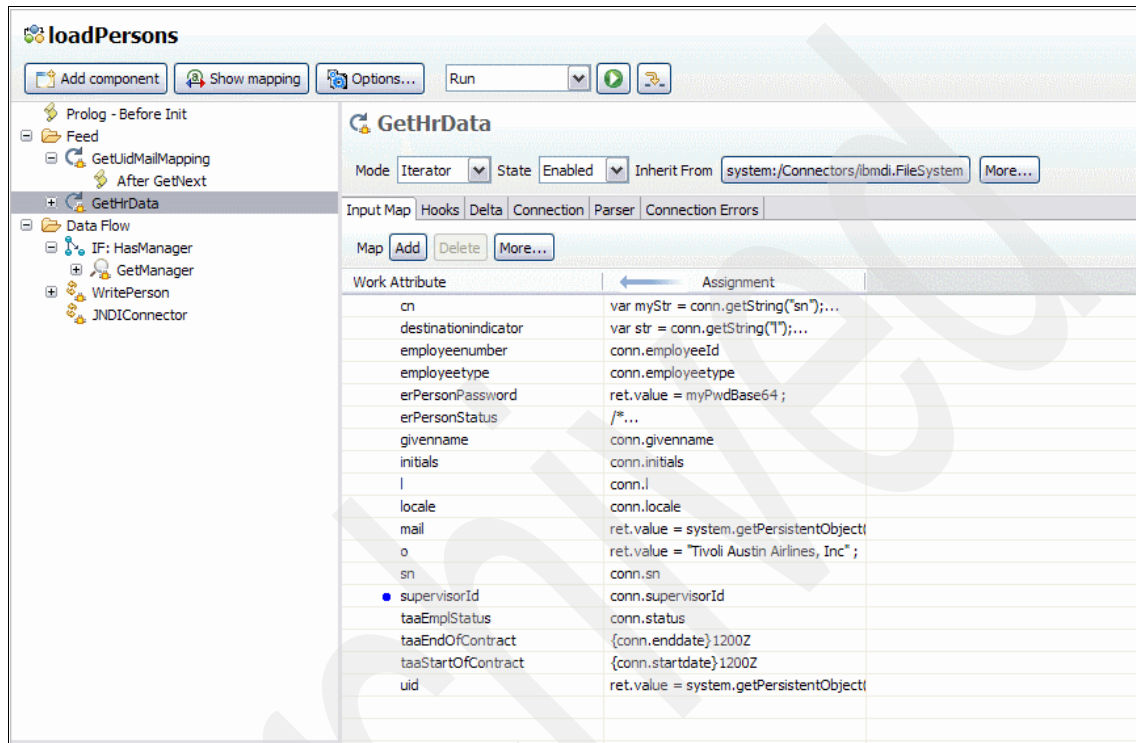


Figure 9-18 Data flow and input map of ongoing identity feed

The components perform the following functions:

GetHrData

Reads the input file. Maps the input file data to attributes on the Directory Integrator work object.

HasManager

Controls the workflow within the AssemblyLine.

GetManager

Searches the Identity Manager LDAP server for a person object having an employee number matching the current input record's supervisorId field. Gets the object's distinguished name.

WritePerson

Creates a new person or updates an existing person in Identity Manager using the data obtained by the other connectors.

The next four sections look at the detailed configuration of each component.

The GetHrData connector

The GetHrData connector is a file system connector with a CSV parser. It runs in iterator mode. Figure 9-18 on page 394 shows the input map for this connector. Most of the attribute mappings simply map the connector attribute to a work attribute of the same name. However, some of the work attributes are set using scripts.

The cn work attribute is built using a script that concatenates the values of givenname, initial, and sn as read from the HR data file. Each value is separated by a single space.

The erPersonStatus work attribute is set using the following JavaScript:

```
/*
 * Convert the TAA HR system's status values into an
 * active/suspended status for Tivoli Identity Manager.
 * 'A' is an active employee. 'L' is someone on a leave
 * of absence. Both of these are considered active in
 * Tivoli Identity Manager. There is also an 'I' status for
 * inactive employee records. This, and any other
 * status, are mapped to a suspended status in Identity
 * Manager.
 * This script assumes that HR Feed entries without 'status' attribute
 * have been skipped before (e.g. at GetNextOK hook)
 * otherwise we will run into an error here
 */
var status = conn.getString('status');
if ( status.equalsIgnoreCase('A') || status.equalsIgnoreCase('L') )
    ret.value = '0'; /* Tivoli Identity Manager's active state */
else
    ret.value = '1'; /* Tivoli Identity Manager's suspended state */
```

This script uses a person's employee status in TAA's human resources system to set the value of the erPersonStatus attribute on the corresponding person record in Identity Manager.

The taaEndOfContract and taaStartOfContract attributes are both built using a simple expression statement that attaches the fixed string "1200Z" to the HR data file values of startdate and enddate. The resulting string has the syntax of <YYYYMMDDssmm> followed by a "Z". This is the date format as required by Tivoli Identity Manager.

The HasManager IF-branch

The IF-branch controls whether it is feasible to search for a person's manager entry in Identity Manager. If the attribute *supervisorId* is missing from the person entry provided by the HR data file, then the AssemblyLine skips the GetManager connector. The IF-branch test condition is configured to check for the existence of the attribute *supervisorId* on the work object.

This is a JNDI connector. It runs in update mode. It connects to the Identity Manager server using a special JNDI service provider that is a part of the Identity Manager server.

The connector issues a lookup request to the Identity Manager JNDI service to see whether the person described by the data in the Directory Integrator work object matches an existing Identity Manager person. If no matching person is found, the connector issues an add request to the Identity Manager JNDI service. If a match is found, the attributes returned by the lookup are compared with the attributes of the Directory Integrator work object. If no work object attributes exist that are different from the person's current attributes, then the connector exits. If there are either work object attributes with different values or work object attributes that do not exist on the Identity Manager person, then the connector issues a modify request to the Identity Manager JNDI service.

The connector must be configured correctly to connect to the Identity Manager server and to be able to find the correct IDI data feed service object in Identity Manager. The connection data for TAA's WritePerson connector is shown in Figure 9-20.

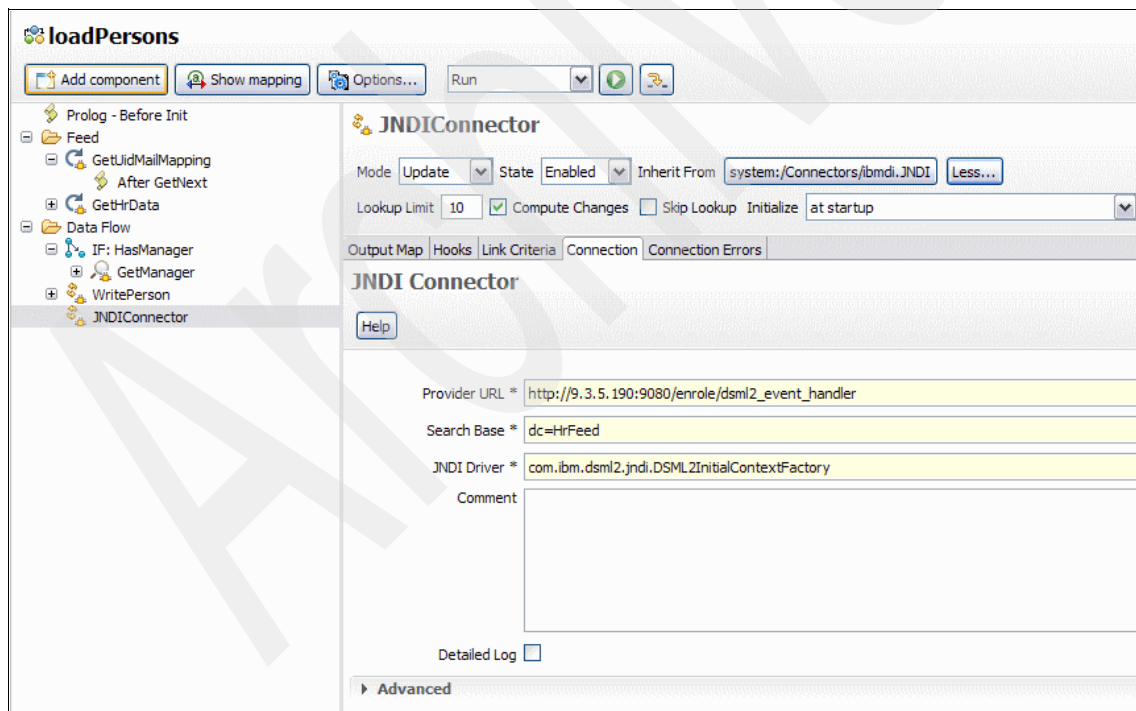


Figure 9-20 Connection data for the WritePerson connector of the identity feed

The JNDI Driver field always contains the same value.

The host name or IP address in the Provider URL field should be the same as what you use when connecting to Identity Manager with a Web browser, but the rest of the URL will always be the same.

The Search Base field must match the value set on the IDI data feed service used by the identity feed. TAA's IDI data feed service is shown in Figure 9-11 on page 382.

Note that the values of these fields have no real meaning. It is only important that the values on the service form match the values on the connector. The user name and password do not need to be a real login name on any platform. They are just used by the Identity Manager JNDI service to authenticate any requests that it receives. The naming context must have a value that meets the syntax requirements of a distinguished name, and the attribute names used in the naming context must be attributes that are defined in the Identity Manager LDAP server's schema. But the naming context does not need to match any existing object in the directory, nor does it matter what values are used in the naming context. It is only used by the Identity Manager JNDI service to determine which IDI data feed service to use when processing a request. For this reason you must ensure that your Identity Manager installation has multiple IDI data feed services and that they have unique naming contexts.

The last field on the connector's configuration tab that you should be aware of is the Search Filter field. This field has no meaning for JNDI connectors running in update mode. The actual search filter used when running the AssemblyLine is build dynamically based on the link criteria. The filter as defined here in the Search Filter parameter is used only if you test your JNDI connection settings from within the mapping table tab by clicking the *connect to data source* icon.

The link criteria used by the Update Tivoli Identity Manager connector is similar to the link criteria of the GetManager connector. It searches for Identity Manager persons having an employee number equal to the value of the Directory Integrator work object's employeenumber attribute.

Directory Integrator loads the attributes of any person found by the link criteria's search into an object named *current*. It is the attributes of the current object that are compared with the attributes of the work object to decide whether an existing user must be modified. "Identity Manager person status" on page 388 discusses how the Identity Manager JNDI service does not return the erPersonStatus attribute in its search results. To avoid unnecessary modify requests and to make sure that the erPersonStatus attribute is maintained through Identity Manager workflows only, the idea is to ignore this attribute in case of JNDI modify operations. This can be achieved by using the Tivoli Directory Integrator feature

to disable modify operations on selected attributes within the connector's output mapping table.

TAA's configuration for the connector's output map is shown in Figure 9-21.

The screenshot shows the configuration interface for the 'loadPersons' connector. The 'WritePerson' connector is selected, and its 'Output Map' tab is active. The table below lists the output map assignments.

Assignment	Add	Mod	Component Attribute
ret.value = "employeeur	true	false	\$dn
work.cn	true	true	cn
work.destinationindicator	true	true	destinationindicator
work.employeenumber	true	false	employeenumber
work.employeetype	true	true	employeetype
work.locale	true	false	erlocale
work.erpersonpassword	true	false	erpersonpassword
work.erPersonStatus	true	false	erpersonstatus
work.givenname	true	true	givenname
work.initials	true	true	initials
work.l	true	true	l
work.mail	true	false	mail
work.manager	true	true	manager
work.o	true	true	o
ret.value = "taaEmployee	true	false	objectclass
work.sn	true	true	sn
work.taaEmplStatus	true	true	taaEmplStatus
work.taaEndOfContract	true	true	taaEndOfContract
work.taaStartOfContract	true	true	taaStartOfContract
work.uid	true	false	uid

Figure 9-21 The WritePerson connector's output map

In most cases the output map simply copies the value of a work attribute to a connector attribute of the same name. The connector attribute names are the names that will be passed to Identity Manager's JNDI service, so they must match the names of attributes in the Identity Manager LDAP schema. Some of the attributes that were read from the input file have different names, so these attributes must be mapped to the correct Identity Manager attribute name. For example, the locale work attribute is mapped to the erlocale connector attribute, and the status work attribute is mapped to the taaEmplStatus connector attribute.

The `objectclass` attribute is always set to `taaEmployee`. This causes the feed to create persons in Identity Manager with TAA's custom Employee entity type. If TAA wanted to use multiple entity types, the feed would need to specify the base object class for whichever entity type is correct for a given input record.

Note: You must ensure that your feed always uses the same object class for a given person. The feed may not attempt to change a person's entity type.

The `$dn` attribute is a special case. The Identity Manager JNDI service expects this attribute to be in the form of a distinguished name where the first element uniquely identifies the person, and the remainder is the naming context of the Identity Manager IDI data feed service. The `$dn` connector attribute is set with a script that combines the value of the unique `$employeenumber` work attribute with the value of the Naming Context field of the connector's configuration tab. The source for this script is:

```
ret.value = "employeenumber=" + work.getString("employeenumber")
           + "," + WritePerson.getConnectorParam("jndiSearchBase");
```

One last point to notice in the output map is that not all attributes are being mapped during modify operations. Any time that you want to set an initial default value with an identity feed but do not allow that value to be changed within Identity Manager, you must exclude that attribute from modify operations. You can see this by the check boxes in the Mod column in Figure 9-21 on page 399.

The `$dn` and `objectclass` attributes are never set in modify operations. The values of these attributes should never change, so it should never be necessary to include them in a modify operation. It is likely to cause errors when you run your feed if you do include them accidentally.

Since the `employeenumber` is supposed to be the unique key of a person, it should never change. Therefore, updates of this attribute are not allowed.

TAA has also decided to exclude the `erlocale` attribute from modify operations. This allows Identity Manager administrators, or the persons themselves, to change their preferred language without having the next identity feed overwrite their choice.

The final configuration step for this connector is to define a script for the *default on error* hook. If the connector encounters any errors, and no error catching hooks are defined, the entire `AssemblyLine` will be aborted. TAA does not want to abort the feed just because of a failure on a single person. The script for this hook simply writes the Directory Integrator error object to the feed's log. The error object contains a description of the error that was caught.

The source of the script is:

```
task.dumpEntry(error);
```

Modifying the identity feed for the initial load

In “Considerations for initial loads” on page 380 we discuss how some of the functionality in the ongoing identity feed is not appropriate to the initial load of existing persons. Specifically, generating unique values for UIDs and e-mail addresses is not appropriate during the initial load because existing employees already have values assigned for both attributes. Those values must not be changed during the initial load. There can also be issues with the order in which persons are created if there are references between the persons, for example, manager-employee relationships. TAA must address both of these issues.

Existing TAA employees have a unique ID that is used as their account name on all platforms. This value, and their e-mail addresses based on this value, must be included on the Identity Manager person objects created by the initial load. These fields will not be maintained by the ongoing feed.

It would be easy to add a lookup connector to the identity feed’s Directory Integrator AssemblyLine if this data were available in a database or directory. TAA does not have the IDs and e-mail addresses available in this form. Instead, they have created a CSV file that maps each employee’s employee number to his ID and e-mail address.

Directory Integrator’s file system connector does not provide the ability to perform searches in a file, so it cannot be used in lookup mode. Directory Integrator does have an embedded database, so the employee number to ID and e-mail address mappings can be loaded into this database before the feed starts the initial load of person data.

Changes are also required to ensure that supervisor relationships are not created until the supervisor’s Identity Manager person object exists. There are multiple ways to ensure that the supervisor’s Identity Manager person object is available before supervisor relationships are being created, such as:

- ▶ Load person entries in a sorted order according to the organizational hierarchy of the enterprise. Beginning at the top management going down the hierarchy levels eliminates the risk of loading employee entries before their manager entry. Of course, requires that the export from the HR database can be done in hierarchical order.

- ▶ Another simple approach is to run the initial load at least twice. The first might fail on the manager lookup for some person entries because the manager has not been loaded into Tivoli Identity Manager yet. In this case the person entry still will be imported, but the manager attribute will be empty and therefore ignored on the JNDI connector's output map. On the second run all manager lookups will run successfully and the missing supervisor relationships will be updated.
- ▶ One more alternative is to set up the initial feed as two separate AssemblyLines. Each AssemblyLine will implement one out of two phases. The first AssemblyLine's job is to read the HR data and the uid-email-mapping file and merge correlating person entries. It will import the person objects into Identity Manager, leaving the manager attribute empty. Then the second AssemblyLine again iterates through the HR data file. It does the manager lookup and updates each Identity Manager person entry to set the manager attribute. All other person attributes will not be changed in this phase.

TAA made the decision to implement the initial HR feed as well as the ongoing feed within one single AssemblyLine. It runs twice during the initial load to tackle the supervisor relationship issue and it reads the uid-email-mapping table as well as the HR data file to be able to merge available data of existing employees when adding new Identity Manager person entries. This approach saves you from developing and testing multiple AssemblyLines.

The AssemblyLine is based on the one described in "Implementing the identity feed using Directory Integrator" on page 393. A new connector, named `GetUIdMailMapping`, is added. It is a file system connector using a CSV parser running in iterator mode to read the file containing the employee number to ID and e-mail address mappings, and store those in Directory Integrator's embedded database before the `GetHrData` connector begins reading its input file.

The mappings are stored in the embedded database by a script in the `GetUIdMailMapping` connector's after `GetNext` hook. This script saves each record returned from the mapping file in the Directory Integrator system store using the employee number as the key. It then calls the `system.skipEntry()` method. This aborts the processing of the remainder of the AssemblyLine for the current record. The result is that the `GetUIdMailMapping` connector is run repeatedly until it runs out of input data. At that point the AssemblyLine moves on to the `GetHrData` connector and processes its records normally.

The source of the GetUIdMailMapping connector's After GetNext hook is:

```
/*
 * Save the uid and email in the system store using
 * employeeNumber as the key. Then skip the rest of
 * the AssemblyLine. These records will be retrieved
 * during the processing of the real HR dump file.
 */
system.setPersistentObject(conn.getString('employeeId'),conn);
system.skipEntry();
```

The GetHrData connector's input map has been modified to look up the person's ID and e-mail address from the embedded database. Figure 9-22 shows the new input map with scripts getting the values of these new attributes.

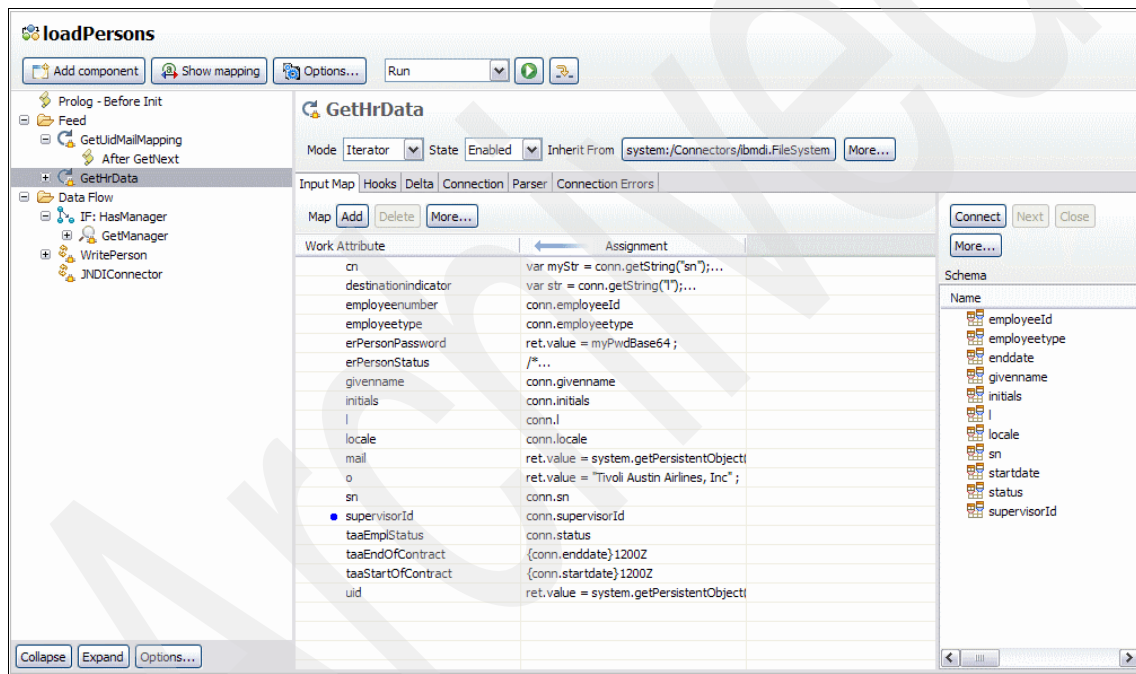


Figure 9-22 Input map setting uid and mail attributes for the initial feed

The new mail work attribute is set by the script:

```
var myUid = system.getPersistentObject(conn.getString("employeeId"));
if (!isNull(myUid))
    ret.value = myUid.getString("mail");
```

The uid attribute is being set by a similar script, except that script gets the uid attribute from the object in the embedded database:

```
var myUid = system.getPersistentObject(conn.getString("employeeId"));
if (!isNull(myUid))
    ret.value = myUid.getString("uid");
```

The one remaining change to this AssemblyLine is to add the two attributes mail and uid to the WritePerson connector's output map having the *modify* option disabled. Deselecting the *modify* option here is a must to meet the requirement not to change both attributes during ongoing HR feeds.

The final output map of the WritePerson connector is displayed in Figure 9-23.

Assignment	Add	Mod	Component Attribute
ret.value = "employeeur	true	false	\$dn
work.cn	true	true	cn
work.destinationindicator	true	true	destinationindicator
work.employeeenumber	true	false	employeeenumber
work.employeetype	true	true	employeetype
work.locale	true	false	erlocale
work.erpersonpassword	true	false	erpersonpassword
work.erPersonStatus	true	false	erpersonstatus
work.givename	true	true	givename
work.initials	true	true	initials
work.l	true	true	l
work.mail	true	false	mail
work.manager	true	true	manager
work.o	true	true	o
ret.value = "taaEmployee	true	false	objectclass
work.sn	true	true	sn
work.taaEmplStatus	true	true	taaEmplStatus
work.taaEndOfContract	true	true	taaEndOfContract
work.taaStartOfContract	true	true	taaStartOfContract
work.uid	true	false	uid

Figure 9-23 WritePerson connector output map

9.7 Reconciliation

Reconciliation is the process of determining the accounts existing at a particular managed resource and processing each against provisioning policies defined within Tivoli Identity Manager based on the owner.

9.7.1 Requirements

There is no explicit functional requirement to specify that this must be done. It is the underlying concept of identity management as a process that defines that this must be a requirement. That is, there must be a mechanism to correlate people and the current state of all their accounts to allow for security policies to be enforced and to meet audit and compliance requirements of the business.

9.7.2 Design considerations

Reconciliation is a process that is mostly controlled by the Tivoli Identity Manager application. However, the administrator of the service that is responsible for the reconciliation process does need to define the following:

- ▶ When scheduled reconciliations must be automatically executed by Tivoli Identity Manager

This decision depends on the number of services defined within the system and how often they must have reconciliations run. For example, a managed resource where changes to accounts do not occur often may only require a weekly reconciliation, while others that change on a daily basis may need daily reconciliations. Care must be taken to ensure that reconciliations do not all run at the same time, as this may cause system resources to be completely consumed. A good approach is to estimate the amount of time each reconciliation within a production environment takes to run based on the user population numbers and the time taken in the test environments. Based on this information, reconciliations can then be scheduled to work together. For example, there may be three reconciliations that must be run daily and each may take two hours to run. The schedules can be set to run two hours apart, one after the other so that each has the benefit of being able to fully utilize the system resources. This can be refined and tuned in a production environment during the course of normal operations. Given the processor utilization of a reconciliation process, they are also typically run after business hours.

- ▶ The attributes returned from a reconciliation

Tivoli Identity Manager allows for the selection of the attributes to be retrieved from the managed resource. By default, Tivoli Identity Manager retrieves all attributes, but the reconciliation definitions allow for the exclusion of attributes on reconciliations. This is typically done in cases where the values of certain attributes do not need to be reflected in Tivoli Identity Manager, as they serve no useful purpose or are not to be checked for compliance against policies.

- ▶ If all accounts returned from a reconciliation are to be checked against provisioning policies or whether Tivoli Identity Manager should only process the changes against provisioning policies

Having Tivoli Identity Manager only process changes shortens the amount of time taken to complete a reconciliation due to the reduced number of policy evaluations. The approach taken is normally to have all accounts processed against policy to ensure that a complete policy evaluation is done during all reconciliations. The decision to only process the changes against policy is usually made for performance reasons. For example, in environments where user population numbers are extremely large and the system resources are not sized to manage these numbers (if there are cost restrictions on hardware, for example), the reconciliation process may not finish in time for the start of the next business day. In these cases, it may be prudent to only process changes during nightly reconciliations, and have the system process all account changes over the weekend when there is a larger window of opportunity to process the accounts. Note that there is an option to set a limit on the amount of time a reconciliation can run for, but this does not ensure that processing is performed on the accounts that must have policy evaluated and, as a result, is not commonly used for environments where it is critical that account compliance is to be checked regularly.

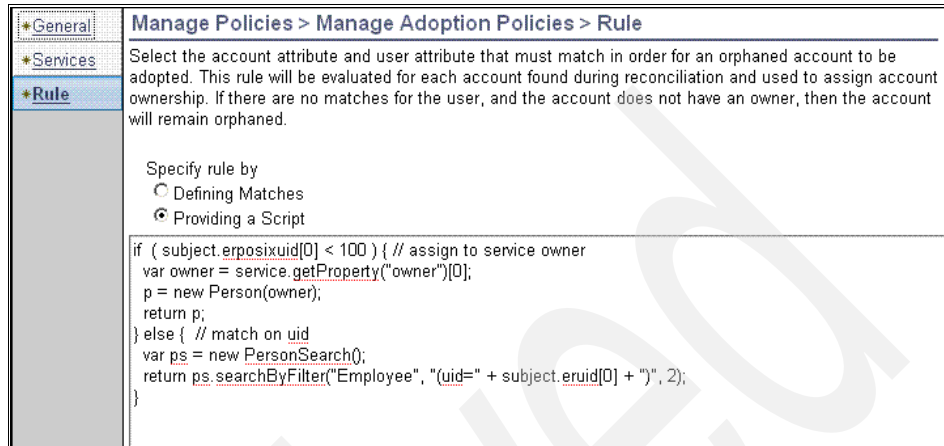
- ▶ How to determine the owner of an account

Reconciliations allow the ability to script the *adoption* process of an account. That is, the rule is a mechanism that allows the reconciliation process to automatically assign an account on the managed resource to a person in Tivoli Identity Manager if it is not known who the owner of the account is. If one cannot be found, then it is deemed to be an orphan. Analysis of all accounts in the environment must be performed to determine a unique and common way to identify accounts and their owners. This then must be scripted and implemented either as a global adoption policy, service type adoption policy (for example, all Windows Active Directory services), or service instance adoption policy (for example, a Windows Active Directory service definition for domain A). The order of precedence of adoption policies is:

- a. Service instance adoption policy
- b. Service type adoption policy
- c. Global adoption policy

The global and service type based adoption rules are defined at Configure System → Global Adoption Policies in Tivoli Identity Manager, while the service instance based adoption rule is defined at Manage Policies → Manage Adoption Policies. Figure 9-24 on page 407 shows an example adoption rule defined within a service instance's reconciliation setting that assigns all Linux accounts where the UID is less than 100 to the Linux service owner (defined in the Tivoli Identity Manager service definition form).

Otherwise, it assigns accounts to people based on their uid attribute matching the Linux account UID.



Manage Policies > Manage Adoption Policies > Rule

Select the account attribute and user attribute that must match in order for an orphaned account to be adopted. This rule will be evaluated for each account found during reconciliation and used to assign account ownership. If there are no matches for the user, and the account does not have an owner, then the account will remain orphaned.

Specify rule by

- Defining Matches
- Providing a Script

```
if ( subject.erposixuid[0] < 100 ) { // assign to service owner
var owner = service.getProperty("owner")[0];
p = new Person(owner);
return p;
} else { // match on uid
var ps = new PersonSearch();
return ps.searchByFilter("Employee", "(uid=" + subject.eruid[0] + ")", 2);
}
```

Figure 9-24 Example adoption rule for a Linux service instance

9.7.3 TAA's implementation

As this is the initial phase of the implementation, TAA has decided to reconcile all account attributes and have policy checked for all accounts returned on a reconciliation. Performance and long-running reconciliations are not deemed to be an issue, as the machines are adequately scaled and the user population is relatively manageable. This leaves the adoption rules and the planning of scheduled reconciliations.

Adoption rules

Where possible, it has been decided to leverage the use of a global adoption rule and specify service profile adoption rules where appropriate. TAA decided to avoid the use of service-instance-specific adoption rules until later stages when requirements are defined to help decide whether these are required.

Figure 9-25 shows TAA's implementation of its global adoption rule that adopts accounts based on its account UID (usually the login ID) matching its Tivoli Identity Manager UID.

The screenshot shows a configuration window titled "Configure System > Global Adoption Policies > Rule". The left sidebar has "General" and "Rule" sections. The main area contains the following text:

Select the account attribute and user attribute that must match in order for an orphaned account to be adopted. This rule will be evaluated for each account found during reconciliation and used to assign account ownership. If there are no matches for the user, and the account does not have an owner, then the account will remain orphaned.

Specify rule by

- Defining Matches
- Providing a Script

```

if ((subject["eruid"]==null){
return null;
} else if (subject["eruid"]!=null){
var buff='{}';
for(i=0;i<subject["eruid"].length;i++){
buff+='uid='+subject["eruid"][i]+'';
}
buff+='}';
var ps = new PersonSearch();
var searchResult = ps.searchByFilter("",buff, 2);
if (searchResult==null && searchResult.length>0)
return searchResult;
else {
return null;
}
}

```

Figure 9-25 TAA global adoption rule

Reconciliation schedules

Table 9-6 shows the scheduled reconciliations for all the services in the TAA production environment.

Table 9-6 TAA reconciliation schedules

Service name	Scheduled reconciliation	Reason
ITIM Service	Not applicable	There is no need to schedule one, as the data is already contained within Tivoli Identity Manager. In fact, there is no option to configure reconciliations for this service.

Service name	Scheduled reconciliation	Reason
Tivoli Access Manager	Daily at 9:00 p.m.	Administration at this stage is still allowed directly through the Tivoli Access Manager administration console and changes occur often, as this is a commonly used piece of infrastructure. Hence, it is TAA's preference to have these reflected in Tivoli Identity Manager often. Reconciliations are expected to not run over an hour due to the manageable user population.
Lotus Notes e-mail	Daily at 10:00 p.m.	Administration at this stage is still allowed directly through the Lotus Notes administration console and changes occur often, as this is a commonly used piece of infrastructure. Hence, it is TAA's preference to have these reflected in Tivoli Identity Manager often. Reconciliations are expected to not run over an hour due to the manageable user population.
Central IT data center RACF	Weekly on Saturdays at 11:00 p.m.	Administration at this stage is still allowed directly through RACF. Administrative actions, however, are not commonly modified. Hence, it has been deemed acceptable to use a weekly reconciliation to update user RACF details in Tivoli Identity Manager. Reconciliations are expected to not run over half an hour due to the smaller user population when compared to Tivoli Access Manager and Lotus Notes. That is, not all staff have an account.

Service name	Scheduled reconciliation	Reason
Central region Linux	Daily starting at 11:30 p.m. with 30 minute delays between each Linux service	Administration at this stage is still allowed directly through Linux and changes occur often, as this is a commonly used piece of infrastructure. Hence, it is TAA's preference to have these reflected in Tivoli Identity Manager often. Reconciliations are expected to not run over half an hour due to the smaller user population when compared to Tivoli Access Manager and Lotus Notes. That is, not all staff have an account.
Corporate HQ Linux		
East region Linux		
West region Linux		
Austin CSC Active Directory	Daily at 1:30 a.m. with 30 minute delays between each Active Directory service	Administration at this stage is still allowed directly through the Windows Active Directory administrative console and changes occur often, as this is a commonly used piece of infrastructure. Hence, it is TAA's preference to have these reflected in Tivoli Identity Manager often. Reconciliations are expected to not run over half an hour due to the fact that users are spread over the CSC domains and hence each domain has a much smaller user population when compared to Tivoli Access Manager and Lotus Notes. That is, not all staff have accounts in the same domain.
Denver CSC Active Directory		
Mexico City CSC Active Directory		
St. Louis CSC Active Directory		
Detroit CSC Active Directory		
New York CSC Active Directory		
Raleigh CSC Active Directory		
Los Angeles CSC Active Directory		
San Francisco CSC Active Directory		
Seattle CSC Active Directory		

Figure 9-26 shows the *Austin CSC Active Directory* service's reconciliation definition. The other reconciliations scheduled to run daily are identical except for the difference in time.


*General	
Manage Services > Set Up Account Reconciliation > Schedule	
Schedule	To schedule a reconciliation for the TAAir Active Directory service service, select the frequency and the time, and then click OK. All times are on the Greenwich Mean Time time-zone.
Query	
	<input checked="" type="radio"/> Daily <input type="radio"/> Weekly <input type="radio"/> Monthly <input type="radio"/> Hourly <input type="radio"/> Annually <input type="radio"/> During a specific month
	At this time 1:30 AM 

Figure 9-26 *Austin CSC Active Directory* service reconciliation definition

Even though the schedules have been designed to avoid timing overlaps between them, there is no reason to do this other than to ensure that each has the maximum amount of system resources to use when required. That is, it is possible to run multiple reconciliations concurrently provided that the resources are available to do so.

When user data is in the system (from the identity feed) and all the reconciliation schedules are defined, Tivoli Identity Manager is ready to have reconciliations run for each service. Depending on the availability of the system and resources, it may be decided that the reconciliations are to be run when they are scheduled. That is, they are deferred to Tivoli Identity Manager to run. TAA has decided to run the initial reconciliation manually to allow population of account data into the system as soon as possible. This must be executed manually for each service. An example of doing this for the Austin Linux service is shown in Figure 9-27 and Figure 9-28 on page 413.

Manage Services > Select a Service

To locate a service that you want to manage, type information about the service in the field, select a service type, and then click Search. The services that match your criteria are displayed in the table below. By default, clicking Search will search the system based on the beginning letters of the item you are searching for. To search for a textual pattern in the middle of an item, use the "*" symbol on the keyboard to indicate a wildcard. (For example, typing "b*" will find "abc".)

Search by
 Search information Service Service type: All

Services

To perform a particular task on a service, click the icon next to the service name, and then select the task you want to perform.

8 results found for: *

Select	Service Name	Description	Service Type	Business Unit
<input type="checkbox"/>	HrFeed	Daily feed of HR data	IDI data feed	Tivoli Austin Airlines
<input type="checkbox"/>	ITIM Service		ITIM	Tivoli Austin Airlines
<input type="checkbox"/>	Local Windows	Local Windows service	Windows Local Account Profile	Tivoli Austin Airlines
<input type="checkbox"/>	PC Procurement Service	Order PC for IT User	PC Procurement	Tivoli Austin Airlines
<input type="checkbox"/>	TAA Access Manager Service		TAM Combo Profile	Tivoli Austin Airlines
<input type="checkbox"/>	TAAir Active Directory service		Active Directory Profile	Tivoli Austin Airlines
<input type="checkbox"/>	TAA LDAP Directory service	Configure Policy Enforcement...	DAP profile	Tivoli Austin Airlines
<input type="checkbox"/>	TAA Linux service	Manage Groups and Access...	POSIX Linux profile	Tivoli Austin Airlines

Page 1 of 1 Total: 8

Context menu for TAA Linux service:

- Change
- Delete
- Set Up Reconciliation
- Configure Policy Enforcement...
- Manage Groups and Access...
- Request Accounts...
- Accounts...
- Account Recertification Status...
- Account Defaults...
- Reconcile Now**

Figure 9-27 Initiate reconciliation manually

Manage Services > Reconcile Now > Select Query

Select or define a query for reconciliation of the **TAA Linux service** service, and then click Submit to perform the reconciliation. The query search string must be a valid LDAP filter. For more information, refer to RFC 2251 documentation for LDAPv3.

Query

- None
- Use query from existing schedule
- Define query

Select	Name	Schedule	Description
<input type="radio"/>	Reconciliation Schedule for TAA Linux service	Daily: Hour:5 Minute:0	Default Reconciliation Created By Create Service Wizard

Page 1 of 1 Total: 1 Displayed: 1 Selected: 0

Figure 9-28 Select query from existing schedule

When the reconciliation task completes, the accounts and their respective owners are reflected in Tivoli Identity Manager. TAA now relies on the automated reconciliation schedules to perform the reconciliations moving forward.

Reports

At the completion of the reconciliation tasks, TAA decided to run the following standard reports to allow for tracking of data produced by reconciliations:

- ▶ Summary of accounts on service
- ▶ Reconciliation statistics

These reports can be found at the Reports → Service Reports window of the Tivoli Identity Manager console, as shown in Figure 9-29. Reports can be extracted either in CSV or PDF format. It is a documented and mandated process for TAA system administrators to run the same reports daily at 10 a.m. and archive these reports to a central repository for use in auditing-related activities.

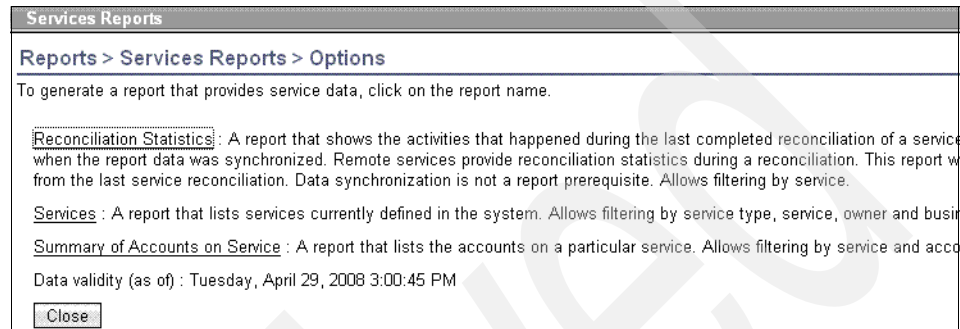


Figure 9-29 TAA reports to run after reconciliations

9.8 Orphan account cleanup

In the context of Tivoli Identity Manager, an orphan account is defined as one that does not have an owner assigned. The existence of an orphan account indicates that it may belong to a person who no longer needs it or is no longer employed by the company, or that the account belongs to someone but has not been automatically adopted via the reconciliation process because the account does not meet the criteria specified within the adoption rules.

9.8.1 Requirements

TAA needs a better way to deal with its audit and compliance issues. Part of this includes addressing the compliance issues by identifying accounts that are not owned by people. That is, there do not exist accountability controls for actions performed using orphan accounts. The process of identifying orphan accounts and performing the relevant corrective actions aims to address these issues.

9.8.2 Design considerations

The initial reconciliation process retrieves all the accounts on all managed resources. Subsequent reconciliation processes may identify new orphan accounts that also must be dealt with in a similar manner. This invariably leaves a trail of orphan accounts throughout the lifetime of any heterogeneous

environment until such a point where the appropriate controls are put in place to prevent the existence of such orphan accounts.

Each account that is not automatically adopted during the Tivoli Identity Manager reconciliation process is classified as an orphan account. Orphan accounts will remain as such until they meet the adoption policy and can be automatically assigned to a person, they are deleted, or they are manually assigned an owner by the relevant system administrator.

To be able to systematically work through the list of orphan accounts, one first needs a list of orphan accounts to begin with. This can be done by generating a report that contains all orphan accounts. By default Identity Manager provides many predefined reports. One of those generates the list of orphan accounts. Figure 9-30 shows how to run the report.

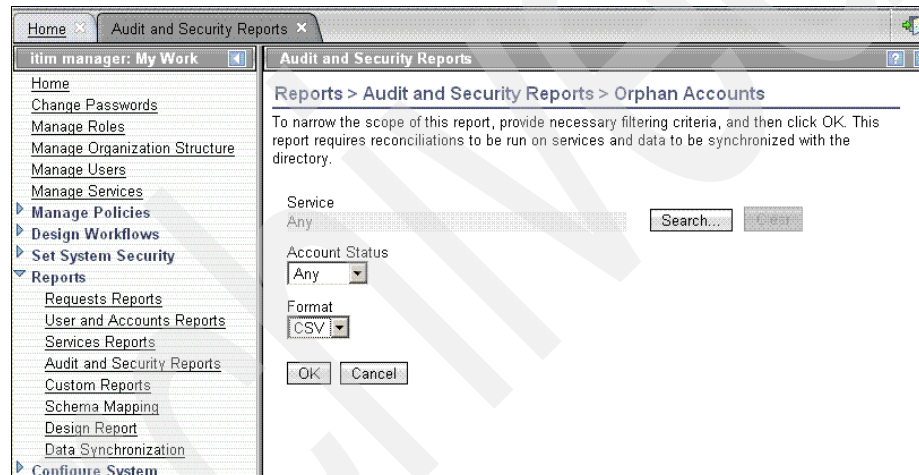


Figure 9-30 Run orphan accounts report

As an alternative, an Identity Manager administrator or the service owner can get to orphan accounts from the services menu directly. Figure 9-31 and Figure 9-32 on page 416 show how to run a search on orphan accounts from the advanced search window and a sample result list.

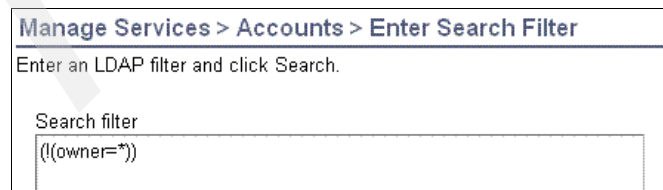
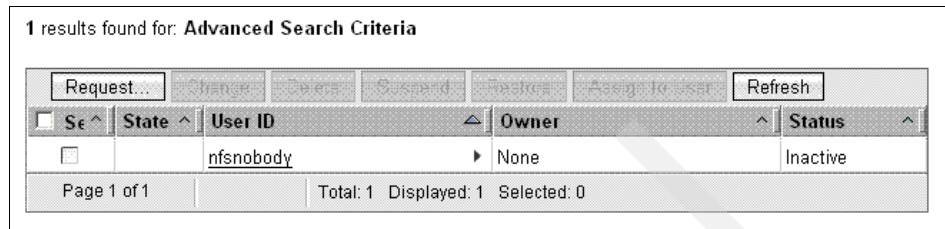


Figure 9-31 Show orphan accounts for service

Running the search brings up the search result window, as shown in Figure 9-32.



1 results found for: **Advanced Search Criteria**

Request...	Change	Delete	Suspend	Reactivate	Assign to user	Refresh
Se ^	State ^	User ID ^	Owner ^	Status ^		
<input type="checkbox"/>		nfsnobody	None	Inactive		

Page 1 of 1 Total: 1 Displayed: 1 Selected: 0

Figure 9-32 Sample orphan account search result list

This can be done for each service instance defined within Tivoli Identity Manager. Each service should have a service owner or a designated person who has the relevant business knowledge and authority required to make decisions about who an account belongs to or to authorize the removal or suspension of the account through Tivoli Identity Manager. Doing so ensures that the audit trail is maintained.

In certain cases, it may be deemed that an account is required but does not have a rightful person who should own it. In these cases, they should be explicitly excluded from reconciliation or assigned to a person object within Tivoli Identity Manager who is not a real person, but simply a *placeholder* entity to own all the accounts that are required but have no rightful owners. System accounts such as root in Linux are good examples of this.

The processing of orphan accounts should be a regular event that is actioned by an administrator or a set of administrators. Where possible, it should be the aim to not have orphan accounts within the Tivoli Identity Manager system.

9.8.3 TAA's implementation

At this stage of the implementation TAA processes orphan accounts manually and uses the following approaches:

- ▶ An administrator is to manually process orphan accounts for all managed resources daily at 11 a.m. by using the Tivoli Identity Manager application Web interface.
- ▶ System owner person entities are created within Tivoli Identity Manager for each managed resource, and all system accounts returned from reconciliations that are not automatically adopted will be manually assigned/adopted to the relevant system owner person entity by a system administrator.

- ▶ Orphan accounts that are not owned by people and cannot be assigned to a system owner person entity are suspended via the Tivoli Identity Manager interface.
- ▶ Accounts that can be assigned to person entities in Tivoli Identity Manager but have not been adopted during reconciliation are manually assigned to the relevant person within Tivoli Identity Manager after confirmation has been received from a relevant business owner of the managed resource.
- ▶ A standard report of orphan accounts is sent to the appropriate business owners to determine whether the account should be de-provisioned or whether there is a rightful owner. Orphan accounts that are suspended and not owned by an Tivoli Identity Manager person entity are subsequently de-provisioned via Tivoli Identity Manager within a week.

Archived

Archived

Technical implementation: Phase II

In this chapter we provide information about how the functional requirements identified in 8.2, “Functional requirements” on page 325, and mapped to this phase are satisfied.

Table 10-1 shows the functional requirements versus the implementation steps for this phase.

Table 10-1 Mapping of functional to deployment requirements

Functional requirement	TAA's deployment requirement
A. Users have a single password for all of their accounts.	1. Common account creation for new hires
B. Password resets are delegated to users other than the system administrators, possibly the end users.	2. Password synchronization using the Windows password interceptor
C. Common values are entered automatically.	1. Common account creation for new hires
D. Manually entered values can be checked for correctness.	1. Common account creation for new hires

Functional requirement	TAA's deployment requirement
J. Automatically create common accounts when a person is employed.	1. Common account creation for new hires
U. An employee's accounts will be disabled or removed when the identity feed shows that an employee has become inactive.	3. Account suspension on termination

10.1 Common account creation

Based on business and functional requirements, the next step to implement is the common account creation in TAA's environment. This section describes the design considerations and TAA's implementation for this functionality.

10.1.1 Requirements

Because of the requirements described in 8.2, "Functional requirements" on page 325, new hires should be provisioned with accounts that they require for their jobs. The provisioning process must be executed automatically.

The provisioning process generates accounts in corporate applications for new users. The process uses the following characteristics:

- ▶ The corporate directory is represented by the data feed for automatic provisioning.
- ▶ Basic personal information is filled into accounts automatically (for example, user name, name, surname, last name, title, and so on).
- ▶ Passwords between accounts will be synchronized.

10.1.2 Design considerations

There are many aspects to consider when trying to create common accounts for users. TAA is planning to automatically create a set of accounts for users that have been hired in order to provide new users with the necessary tools to perform their duties faster. These aspects are discussed in this section.

Services

Services define what adapters are being managed by Tivoli Identity Manager. Each service profile (or adapter type) has its own characteristics and requires specific parameters.

Adapters should not be completely tested until several additional configurations are completed. Reconciliation should only be executed once the services are defined in their correct place in the organization tree. Services cannot be transferred, so if all the accounts associated with a particular service must be deleted, then the service is deleted as well. This can be avoided only by performing a reconciliation after the service is placed in its definitive place. Reconciliation also yields better results if the identities are already created and their aliases are defined correctly. This allows the reconciliation process to associate the identities to the accounts.

In order to create a practical service design, one must know the answers to the following questions:

- ▶ What are the target systems?

This is the first information to be obtained. The list of intended systems is good enough, but there may be special considerations as to how Tivoli Identity Manager adapters are deployed on particular platforms.

- ▶ How are identities assigned to services?

While most services are unique in an organization, some systems have users defined according to location. For example, each site may have its own Microsoft Windows Active Directory domain, and users should be provisioned to the site in which they work.

- ▶ How are the services administrated?

Services can be centrally administrated or have distributed administration. Each method of administration may impose certain requirements in the service design.

Service selection policies

Service selection policies extend the ability of provisioning policies by providing the ability to provision accounts based on person attributes. In order for a service selection policy to be enforced, a provisioning policy must have the service selection policy as its target. The service selection policy then identifies the service type to target and defines provisioning based on JavaScript code.

Service placement

A service can be placed anywhere in the organization tree. This section describes some of the alternatives. There is no reason to limit your design to those shown here.

All services in organization tree root

Technically, all services could be placed at the top level of your organization tree. One of this design's advantages is that all services and provisioning policies can be found easily.

Placing services in admin domains

Admin domains can be used for several reasons. One interesting aspect of these domains is that several identities can be assigned as domain administrators. These domain administrators have many rights on the services and other objects in that domain by default. This can make delegated administration much simpler.

Distribute services throughout the organization tree

Services should be placed in the tree for two reasons: administration and service selection policies. If service selection policies should assign users to the nearest service, having the service on different containers in the tree, near to the users, makes the policy simpler.

Administration of the services and identities may require that services be placed in a container on the tree.

Provisioning policies

Provisioning policies define how users should be provisioned to the systems, or services, as they are called in Tivoli Identity Manager. Each provisioning policy has the following information:

- ▶ General information

Several pieces of information about the policy, including:

- The scope of the policy defines whether only services of the current container or all services in the organization tree are affected.
- A numeric priority value that defines which policy is more important (the lower the number, the higher the priority).

- ▶ Membership

The Membership tab determines who is governed by the provisioning policy.

- ▶ Entitlements

On the Entitlements tab you control a list of provisioning parameters that are applied to each account as it is provisioned to a user. The values that are available vary depending on the managed resource. The Entitlements tab controls how provisioning takes place.

Figure 10-1 shows the typical components of a provisioning policy. The roles referenced by this policy are depicted on the left side of the picture and the target systems on the right side. In the middle are the provisioning policy and a service selection policy.

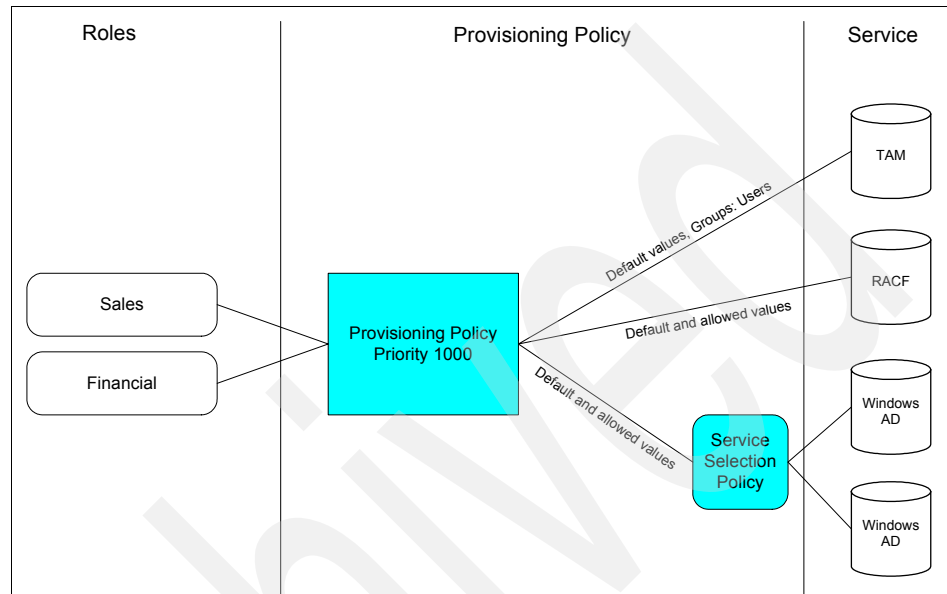


Figure 10-1 Provisioning policies and service selection

When defining a policy, there are two special roles that can be used:

- All** This role applies to every identity in the organization.
- Others** This role applies to people who are not members in any provisioning policy that grants them any of the entitlements in the current policy where others is the membership type. This membership type can be used to create comprehensive policies that grant entitlements to all people in an organization who have not been granted that entitlement through another provisioning policy's membership.

Another element shown in Figure 10-1 is the service selection policy. They extend provisioning policies by providing the ability to provision accounts based on person attributes depending on some business or administrative purpose. For example, if a separate Windows Active Directory domain is deployed in every location, user accounts can be created for the domain closest to the user in the org tree.

Provisioning policy joins

When an account is being created for an identity, the characteristics it has on the target system are determined by the entitlements in the provisioning policies. So if a given user can be provisioned to a system by membership in two or more different provisioning policies, the actual attributes on the system are determined by joining all provisioning policies.

Figure 10-2 shows three provisioning policies that apply to the same service. Each policy has a different priority. Two attributes are determined by each policy (shown on the lines from the policies to the service: login script and groups). In the configuration for these attributes it is stated that groups are determined by *union* and that the login script is determined by *priority*.

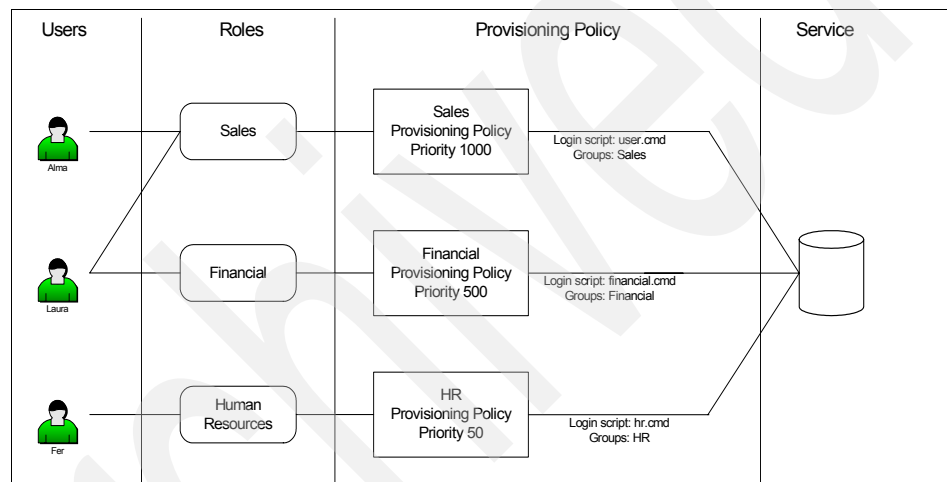


Figure 10-2 Provisioning policy joins

Three users are being provisioned by these policies to the target system. The attributes that each user is assigned on the systems are shown in Table 10-2.

Table 10-2 Result of the provisioning policy join

Identity	Login script Join directive: Priority	Groups Join directive: Union
Alma	user.cmd Since Alma is a member of the sales organizational role she gets provisioned by the sales provisioning policy.	Sales
Laura	financial.cmd Both sales and financial polices apply because Laura is a member of the sales and financial organizational roles. However, she gets provisioned by the financial provisioning policy because it carries a lower priority value.	Sales Financial
Fer	hr.cmd Since Fer is a member of the HR organizational role he gets provisioned by the HR provisioning policy.	HR

10.1.3 TAA's implementation

The first part of this phase includes common account creation, which means that users who are being hired are provisioned with default accounts in an automatic way. Since there is a list of services that TAA's employees must have access to, it is important to decide where they should be located within TAA's organization tree.

There are two services to which all users must have immediate access. They are:

- ▶ Lotus Notes, for e-mail and collaboration
- ▶ Tivoli Access Manager, for access to Web applications

Figure 10-3 on page 427 depicts the design for the service placement within TAA's environment.

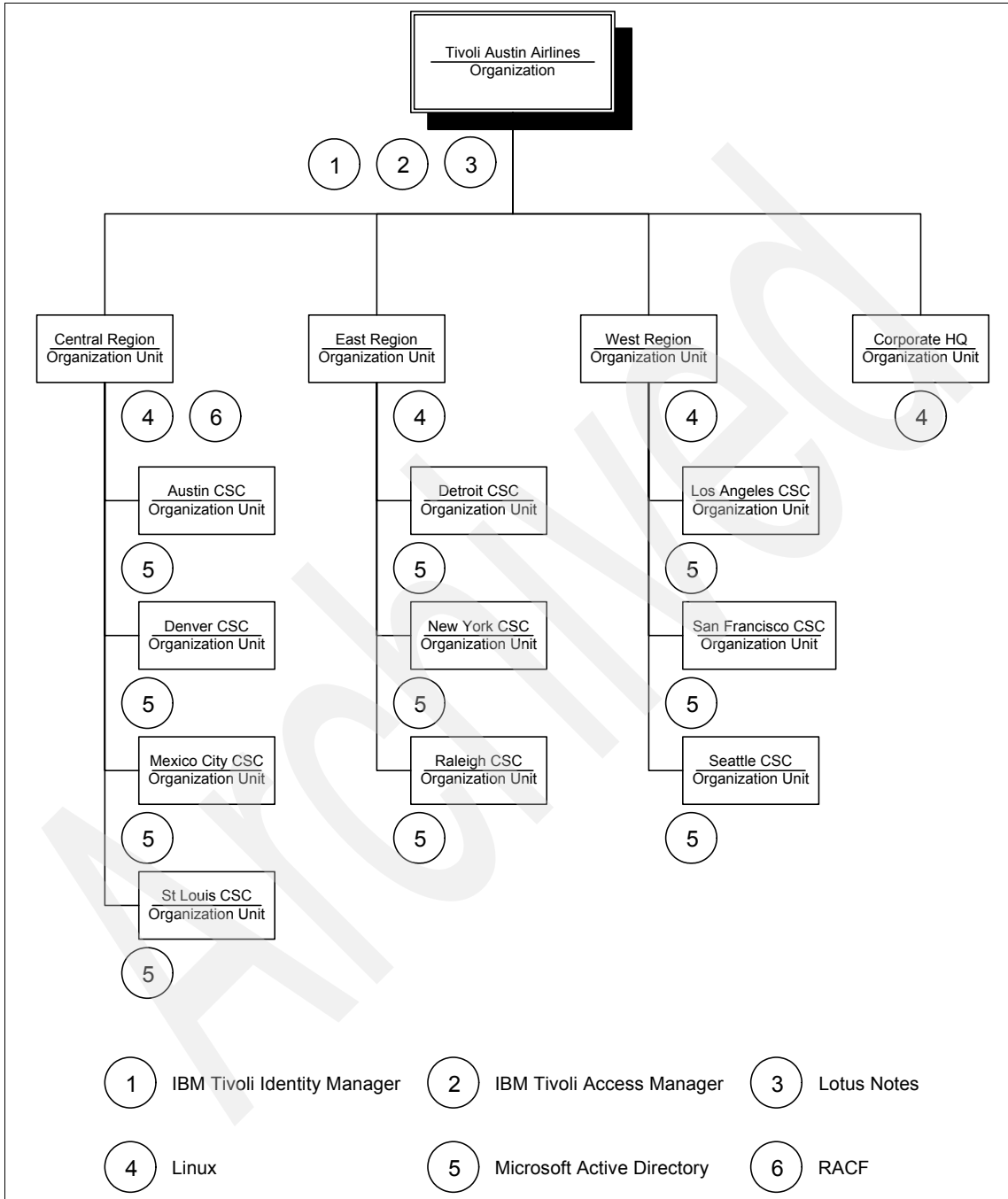


Figure 10-3 Services location within TAA's environment

Lotus Notes and Tivoli Access Manager accounts

First, we must provision the user with Lotus Notes and Tivoli Access Manager accounts automatically. All TAA employees require both accesses to perform their duties. We used the entitlements discussed in “Provisioning policy with Tivoli Access Manager entitlements” on page 283 in an automatic provisioning policy at the top level of the organization tree. The password delivery process is discussed in “Password delivery process” on page 438.

Table 10-3 shows the values used for provisioning Tivoli Access Manager accounts.

Table 10-3 Entitlements

Field	Value
ertam4dn	{"cn="+subject.getProperty("cn")[0]+"",o=users,o=taa,c=users"}
ertam4singesign	TRUE
User Id	{subject.getProperty("cn")[0].toLowerCase().substring(0,1)+subject.getProperty("sn")[0].toLowerCase()}
Last Name	{subject.getProperty("sn")[0]}
Full Name	{subject.getProperty("cn")[0]}

Note: Table 10-3 represents an example of how to automatically fill values for accounts on provisioning policies. For each service type, you must analyze which values accounts will need and create appropriate JavaScripts to perform these operations. For more information refer to the IBM Tivoli Identity Manager Information Center Version 5.1, at:

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

An account will be created automatically with definitions shown in Table 10-3. For example, looking at the information for a specific user in the corporate directory, shown in Table 10-4 on page 429, accounts will be provisioned for Tivoli Access Manager and Lotus Notes.

Table 10-4 Common person attributes

Field	Value
First Name	Alma
Last Name	Ferman
Full Name	Alma Ferman
Employee Number	11111111
Title	Sales Executive
E-mail address	alma.ferman@taair.com
Telephone Number	555 5555

There also exists a provisioning policy for Tivoli Identity Manager accounts. It is created by the installation process, but at this time we only provision users manually.

In phase 3 we enable automatic provisioning of Tivoli Identity Manager accounts to grant all users access to the Tivoli Identity Manager user interface for self-service tasks. Figure 10-4 shows the definition of provisioning policies.

Manage Provisioning Policies

Manage Policies > Manage Provisioning Policies > Manage Provisioning Policy

To locate a provisioning policy that you want to manage, type the information about the policy in the field and click Search. The policies that match your criteria are displayed in the table below. By default, clicking Search will search the system based on the beginning letters of the item you are searching for. To search for a textual pattern in the middle of an item, use the "*" symbol on the keyboard to indicate a wildcard. (For example, typing "b*" will find "abc".)

Search information

Search by
 Provisioning Policy
 Business unit

Provisioning Policies

You can create, change, or delete a provisioning policy. Select the policy, and then click the appropriate button to select the task you want to perform.

18 results found for: *

<input type="checkbox"/> Select	Provisioning Policy	Description	Status	Priority	Business Un
<input type="checkbox"/>	Allow All Accounts (Phase I)	Build for phase I Give Access to all Services.	Enabled	1	Tivoli Austin Airlines
<input type="checkbox"/>	Default provisioning policy for ITIM	Allow everyone to be provisioned for an ITIM account.	Enabled	10000000	Tivoli Austin Airlines
<input type="checkbox"/>	Default Provisioning Policy for service Linux.CentralRegion	Created during service creation	Enabled	10000000	CenterRegion
<input type="checkbox"/>	Default Provisioning Policy for service Linux.CorporateHQ	Created during service creation	Enabled	10000000	AD_CorpHQ
<input type="checkbox"/>	Default Provisioning Policy for service Linux.EastRegion	Created during service creation	Enabled	10000000	EastRegion
<input type="checkbox"/>	Default Provisioning Policy for service Linux.WestRegion	Created during service creation	Enabled	10000000	WestRegion
<input type="checkbox"/>	Default Provisioning Policy for service MS.AD.AustinCSC	Created during service creation	Enabled	10000000	AustinCSC
<input type="checkbox"/>	Default Provisioning Policy for service MS.AD.DenverCSC	Created during service creation	Enabled	10000000	DenverCSC
<input type="checkbox"/>	Default Provisioning Policy for service MS.AD.DetroitCSC	Created during service creation	Enabled	10000000	DetroitCSC

Figure 10-4 Provisioning policies

Both entitlements (Lotus Notes and Tivoli Access Manager) are set to automatic, and there is no approval workflow process because TAA's employees will not have access to the Tivoli Identity Manager user interface until phase 3.

Microsoft Active Directory and Linux accounts

For Microsoft Active Directory and Linux accounts, there are several services located within TAA's organization tree. The main reason for this is that each location has its own Microsoft AD domain and local Linux server. There is a

service selection policy to provision the person with the closest service to the user in the organization tree. This way, depending on where the person is located in the organization tree, it gets access to local resources (Microsoft Active Directory and Linux).

Example 10-1 shows the service selection policy that assigns the appropriate service for that user. This policy works effectively because of the way that services were created, since each organization unit has its own services defined for the local user.

Example 10-1 TAA's basic service selection policy

```
function selectService() {
    var serviceInstance = null;

    serviceInstance = ServiceSearch.searchForClosestToPerson(subject)[0];

    return serviceInstance;
}
return selectService();
```

10.2 Password policy

In this section we discuss the requirements, design considerations, and implementation of a common password policy for accounts on all managed services.

10.2.1 Requirements

TAA wants a global password policy for all of its Tivoli Identity Manager managed services (that is, servers and applications), as follows:

- ▶ Passwords must be changed every 60 days or they automatically expire.
- ▶ Help desk administrators reset passwords, so they require a password entry attribute on their Web page. Passwords are not automatically generated.
- ▶ Following new account creation or a password reset, the account owner has up to four hours to log on using the new password before it automatically expires. During the initial logon Tivoli Identity Manager forces the user to change her password.
- ▶ The maximum number of consecutive unsuccessful logon attempts before the account is suspended is six.

10.2.2 Design considerations

Tivoli Identity Manager contains the features needed to satisfy the requirements.

The password policy settings should be driven by the enterprise's security policies.

10.2.3 TAA's implementation

From the Tivoli Identity Manager Web Administration home page, select the Configuration tab, then select the Properties tab. Enter the password rules specified in Table 10-5.

Table 10-5 Password rules

Rule	Value
Lost password behavior.	Reset password.
Enable password editing.	Yes. (The check box contains a check mark.)
Enable password synchronization.	Yes. (The check box contains a check mark.)
Password expiration period (days).	60
Password retrieval expiration period (hours).	4
Maximum number of invalid logon attempts.	6
Policy enforcement.	Mark

Click the **Apply Changes** button to save this information. See Figure 10-5 for the results.

Set Security Properties

System Security > Set Security Properties

Specify system settings in the sections below.

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: 4

Identity Manager Login Account Settings

- *Identity account password expiration period in days: 60
- *Maximum number of incorrect login attempts: 6

Group Settings

- Automatically populate identity manager groups

OK Cancel

Figure 10-5 Configuration properties

10.3 Password strength policy

This section discusses the requirements, design considerations, and implementation of a common password strength policy for accounts on all managed services.

10.3.1 Requirements

TAA wants a global password strength policy for all of its servers and applications.

10.3.2 Design considerations

The password strength policy should be global in that the rules defined within the policy apply to all of the managed service instances. The Tivoli Identity Manager administrator must consult with the systems administrators for all managed services instances in order to identify a common set of rules. For example, the password length must be adjusted to the lowest common denominator. This means that if there are managed services that cannot process passwords longer than eight characters, the maximum password length in the password policy is eight.

Another example is that not all managed services permit the use of special characters, so the password policy must exclude the use of special characters.

The password policy should reject passwords found in a dictionary and the use of repeating characters should be limited.

10.3.3 TAA's implementation

From the Tivoli Identity Manager Web Administration home page, perform these steps:

1. Select the **Manage Policies** menu.
2. Click the **Manage Password Policies** link.
3. Click the **Create** button.
4. On the **General** tab, provide the attribute values shown in Table 10-6.

Table 10-6 Password strength general attributes

Attribute	Value
Policy name	global_password_policy
Policy caption	global_password_policy
Make policy available to services in:	This business unit and its subunits
Description	Global password strength policy for accounts on all services
Keywords	global password policy
Status	Enabled

5. Click the **Targets** tab, then check the **All Types of Services** check box.

6. Select the **Rules** tab and enter the values specified in Table 10-7.

Table 10-7 Password strength rules

Rule	Value
Minimum length	6
Maximum length	8
Maximum repeated characters	2
Minimum unique characters required	4
Minimum alphabetic characters required	2
Minimum numeric characters required	2
Invalid characters	~'!@#\$\$%^&*()_+ -= {} []\:"';<>? ,./
Required characters	
Restricted characters	
Starts with character	
Repeated history length	6
Reversed history length	6
Disallow user name?	Yes (The check box contains a check mark.)
Disallow user name (with case-insensitivity)?	Yes (The check box contains a check mark.)
Disallow user ID?	Yes (The check box contains a check mark.)
Disallow user ID (with case-insensitivity)?	Yes (The check box contains a check mark.)
Disallow in dictionary?	Yes (The check box contains a check mark.)

- Click the **Submit** button at the bottom of the dialog to save this information. See the results in Figure 10-6.

Manage Policies > Manage Password Policies > Rules

To define the rules for the password policy, type the settings for the password rules that you want to define, and then click OK.

Password Rule	Setting
Minimum length	6
Maximum length	8
Maximum repeated characters	2
Minimum unique characters	4
Minimum alphabetic characters	1
Minimum numeric characters	1
Characters not allowed	
Required characters	
Restricted to characters	
Starts with characters	
Repeated history length	6
Reversed history length	6
Disallow user name	<input checked="" type="checkbox"/>
Disallow user name(with Case-Insensitivity)	<input checked="" type="checkbox"/>
Disallow user ID	<input checked="" type="checkbox"/>
Disallow user ID(with Case-Insensitivity)	<input checked="" type="checkbox"/>

Page 1 of 1 Total: 16 Displayed: 16

OK Apply Cancel

Figure 10-6 Password strength rules

10.4 Password synchronization using the Windows password interceptor

During the first two phases of the technical implementation, users do not access the Tivoli Identity Manager user interface because it would be necessary to set up a basic training for users. For this reason, the password attribute is synchronized and reset by the current user interface, the Microsoft Windows operating system.

10.4.1 Requirements

The TAA user desktops are Windows based. All users log in to Microsoft Active Directory in order to start a session and access corporate applications. Since password reset has become a problem because of the amount of users who forget their passwords, and with the implementation of the new security policy for passwords, reset password is becoming a very hard task. Now that TAA has implemented password synchronization between corporate applications, users just must remember one password.

In case a user must change a password, TAA wants the user to change the password on her own using the Microsoft Windows user interface because the user already knows how to do it and Windows is the first point of entry into all TAA systems and applications.

Since Tivoli Identity Manager synchronizes passwords between applications, it should be the interface for all password changes. But this implies training final users to use Tivoli Identity Manager GUI. On the other hand, if one application interface is used to reset a password, it will not be synchronized with other applications, because the reconciliation process does not reconcile passwords. For the first user self-care approach, password synchronization takes advantage of the Microsoft platform and delegates those passwords resets to users.

10.4.2 Design considerations

Because of the nature of password information, there are many concerns about handling this information.

Password policies

Tivoli Identity Manager can establish password policies to the managed system when passwords are managed through the Tivoli Identity Manager user interface. Password policies should be consistent with password policies in managed systems, because a password creation through Tivoli Identity Manager could fail when being set in the managed system because of the password policies in that system. For managed systems password policy information, refer to the applicable product documentation.

To meet password synchronization requirements, password policies between all managed systems must be consistent. One key attribute that must be handled by automated provisioning policies is the password. Passwords must be generated to match the password policies for the services to which they are provisioning. This may be a challenge if there are several password policies in place.

Password delivery process

Since there are many options to create a password, there are also many ways to deliver passwords to users, for example:

- ▶ Delivering physically through hard copy.
- ▶ Sending password by e-mail to a user.
- ▶ Post it into a URL with SSL enabled (link should be sent to user by e-mail).
- ▶ User could determine it by a simple process (for example, surname plus first four digits of employee number).
- ▶ User could contact the help desk to obtain his initial password.

10.4.3 TAA's implementation

The password generation is done randomly by Tivoli Identity Manager when the person is created. When a new TAA employee arrives, there is a need to deliver the initial password to this new employee. The human resources department, as a part of the new starter pack, notifies the user that he must notify the help desk to obtain his new password. The help desk confirms the user's identity by asking questions about his identity and then chooses an initial password for the new employee and performs a password reset with this new initial password.

Password synchronization

Password synchronization has been enabled in Tivoli Identity Manager, which means that any password change performed in the Tivoli Identity Manager console is reflected in all systems managed by this. As we have discussed, users do not have access to the Tivoli Identity Manager user interface until phase 3. In order to meet the password synchronization requirement, TAA has deployed in this phase the Tivoli Identity Manager Password Synchronization for Active Directory Plug-in on each Microsoft Active Directory domain server in order to catch all password changes made through the Microsoft Windows user interface, send them to the Tivoli Identity Manager Server, and synchronize all accounts for users that have changed their passwords.

The password synchronization plug-in intercepts the domain user password changes and communicates with Tivoli Identity Manager for password rules verification and synchronization. The new password is synchronized with other accounts managed by Tivoli Identity Manager for the domain user. For more information about how to set up this component, refer to the *Tivoli Identity Manager Password Synchronization for Active Directory Plug-in Installation and Configuration Guide Version 5.1, SC23-9622*.

The password synchronization plug-in has been deployed on all of TAA's Microsoft Active Directory domain controllers. Based on the information provided by the *Tivoli Identity Manager Password Synchronization for Active Directory Plug-in Installation and Configuration Guide Version 5.1, SC23-9622*, TAA has installed and configured all password synchronization plug-ins. Figure 10-7 shows the information of one of the password synchronization plug-ins.

The screenshot shows a configuration window titled "IBM Tivoli Identity Manager Password Change Notification Configuration". The window contains the following fields and options:

- Installation Path: C:\Tivoli\PasswordSynch
- ITIM Host Name or IP: taatimserver
- SSL Port Number: 9443
- Service DN: Configure Target Services (button)
- ITIM Principal: itim manager
- Password: [masked]
- Verify password: [masked]
- Max Notify Thread Count: 10
- Agent Host Machine: \\taawinad
- Agent Name: ADAGENT
- Enable Password Synchronization (checked)
- Enable Password Rules Verification (checked)
- Require ITIM Response (checked)
- Enable Logging (checked)
- Connect To Windows Active Directory Adapter Registry (checked)
- Buttons: Cancel, OK

Figure 10-7 Password synchronization plug-in configuration

The installation procedure asks for the information in Table 10-8, and we fill it with those values.

Table 10-8 Configuration values for password synchronization plug-in

Field	Explanation	TAA's values
Installation path	Specifies the installation path for the password synchronization plug-in. The value specified must match with the installation directory value entered earlier in the installation process.	C:\Tivoli\Password Synch
Tivoli Identity Manager Host Name or IP	Specifies the IP address for the Tivoli Identity Manager server.	taatimserver
SSL Port Number	Specifies the SSL port for the Tivoli Identity Manager server. The default SSL port for WebSphere Application Server is 9443 on a single server setup. If you have a WebSphere Application Server cluster, the IBM HTTP Server needs to be configured for SSL. The default port for HTTP SSL is 443. For example, shreth.tivlab.austin.ibm.com:9443	9443
Tivoli Identity Manager Principal	Specifies the Tivoli Identity Manager account under which the password change requests are submitted. The account must have the proper authority to submit password change requests for the desired people. This authority is granted when you create the access control information (ACI) for the Principal account by granting read and write permissions to all the attributes that were listed.	itim manager
Password	Specifies the password for the Tivoli Identity Manager account under which the password change requests are submitted.	Password for itim manager
Verify Password	Specifies the verification field for the Tivoli Identity Manager account password.	Password for itim manager

Field	Explanation	TAA's values
Agent Host Machine	Specifies the name of the computer where the Windows Active Directory Adapter is installed and running, for example, \\mymachine.	\\taawinad
Agent Name	Specifies the adapter's registry key name. This value is ADAGENT.	ADAGENT
Max Notify Thread Count	Specifies the maximum number of password change requests that can be processed by the plug-in at any one time. The plug-in processes password synchronization requests in a multi-threaded manner. This value limits the number of threads to be created so that requests can be processed in parallel.	10
Enable Password Synchronization	Specifies whether password synchronization should be enabled or disabled. When password synchronization is enabled, all password change requests are sent to Tivoli Identity Manager in order to synchronize all passwords affected by the change request. When password synchronization is not enabled, the password synchronization plug-in ignores all password change requests on the managed resource.	Yes (check box)
Enable Password Rules Verification	Validates that the password complies with the password rules defined for the user. When this option is selected, the new password is checked against the password policy rules defined for each account type to be synchronized. Unless the password is valid for all accounts, the password change fails with an error indicating that the new password does not meet specified password rules.	Yes (check box)

Field	Explanation	TAA's values
Require Tivoli Identity Manager Response	This option is enabled only if <i>enable password rules verification</i> is selected. When this option is selected, passwords cannot be changed on Active Directory if Tivoli Identity Manager is unavailable.	Yes (check box)
Enable Logging	Allows administrators to enable logging for password change requests sent to the Active Directory Server.	Yes (check box)

Important: Note that if you enable *password rule verification* and require a Tivoli Identity Manager response, make sure that password policies between Tivoli Identity Manager and Microsoft Active Directory have no conflicts. Otherwise, password reset becomes impossible because Tivoli Identity Manager or Microsoft Active Directory do not allow the password change.

For the service DN value, each password synchronization plug-in contains information about the mapping between the Microsoft Active Directory domain and the Tivoli Identity Manager services configured previously. Table 10-9 shows the information configured.

Table 10-9 TAA's target services

Base point	Service target DN
ou=AustinCSC,dc=taawin2003ad,dc=taa,dc=com	erservicename=Austin CSC Active Directory,ou=Austin CSC,ou=Central Region,o=Tivoli Austin Airlines,ou=taa,dc=com
ou=DenverCSC,dc=taawin2003ad,dc=taa,dc=com	erservicename=Denver CSC Active Directory,ou=Denver CSC,ou=Central Region,o=Tivoli Austin Airlines,ou=taa,dc=com
ou=DetroitCSC,dc=taawin2003ad,dc=taa,dc=com	erservicename=Detroit CSC Active Directory,ou=Detroit CSC,ou=East Region,o=Tivoli Austin Airlines,ou=taa,dc=com
ou=LosAngelesCSC,dc=taawin2003ad,dc=taa,dc=com	erservicename=Los Angeles CSC Active Directory,ou=Los Angeles CSC,ou=West Region,o=Tivoli Austin Airlines,ou=taa,dc=com

Base point	Service target DN
ou=MexicoCityCSC,dc=taawin2003ad,dc=taa,dc=com	erservicename=Mexico City CSC Active Directory,ou=Mexico City CSC,ou=Central Region,o=Tivoli Austin Airlines,ou=taa,dc=com
ou=NewYorkCSC,dc=taawin2003ad,dc=taa,dc=com	erservicename=New York CSC Active Directory,ou=New York CSC,ou=East Region,o=Tivoli Austin Airlines,ou=taa,dc=com
ou=RaleighCSC,dc=taawin2003ad,dc=taa,dc=com	erservicename=Raleigh CSC Active Directory,ou=Raleigh CSC,ou=East Region,o=Tivoli Austin Airlines,ou=taa,dc=com
ou=SanFranciscoCSC,dc=taawin2003ad,dc=taa,dc=com	erservicename=San Francisco CSC Active Directory,ou=San Francisco CSC,ou=West Region,o=Tivoli Austin Airlines,ou=taa,dc=com
ou=SeattleCSC,dc=taawin2003ad,dc=taa,dc=com	erservicename=Seattle CSC Active Directory,ou=Seattle CSC,ou=West Region,o=Tivoli Austin Airlines,ou=taa,dc=com
ou=StLouisCSC,dc=taawin2003ad,dc=taa,dc=com	erservicename=St Louis CSC Active Directory,ou=St Louis CSC,ou=Central Region,o=Tivoli Austin Airlines,ou=taa,dc=com

The service target DN parameter indicates the DN of the service where the person is located in Tivoli Identity Manager. With this mapping, when the password synchronization plug-in intercepts a password change it locates the user in Tivoli Identity Manager and sends the password change.

10.5 Account suspension on termination

It is important to take control of employee accounts during their entire life cycle, from creation (when they are hired) to deletion (when they leave the company). Also, in order to be compliant with regulation and standards, TAA wants to manage in an automatic process the suspension of all accounts associated with a user that leaves the company immediately and then delete the accounts after three months.

Note: TAA is concerned that they may lose important data stored on their Linux systems when automatically deleting an employee's Linux account. They request that the Linux service owner needs to approve the final deletion of accounts on that managed resource. Appendix A, "Account management workflow customization" on page 615 shows a customized workflow of the delete account functionality and how it is used in a related scenario.

10.5.1 Requirements

In a recent IT security audit, as part of TAA's efforts to comply with legal regulations (as described in Chapter 1, "Business context for identity and credential management" on page 3), results show that there are active accounts for users that are no longer TAA employees.

In order to manage the full life cycle of user accounts, the solution suspends all accounts from users that are no longer TAA employees.

The human resources user data directory is the user directory that has the most recent information about TAA employees. As seen in 10.1, "Common account creation" on page 421, the corporate users directory (HR directory) is the feed for user provisioning. Since it is the most accurate and valid user data, it should also be the feed to verify whether users remain TAA employees.

10.5.2 Design considerations

As discussed for phase 1 in 9.6.2, "Design considerations" on page 368, and the implementation of that discussion in TAA's environment (9.6.3, "TAA's implementation" on page 380), TAA employees now are provisioned with default accounts to perform their basic job duties.

In order to keep the Tivoli Identity Manager user database as accurate as possible, a user that has been suspended on the HR user database should also be suspended on Tivoli Identity Manager, and all accounts associated with that user also, because keeping them active will become a security risk since they are not to perform anymore job-related activities.

The account suspension process will be managed by Tivoli Identity Manager. There are some points to consider before implementing this process. The most important is that since this process will be executed in an automatic way, you should first test all conditions, because the process will suspend all user accounts based on one parameter.

10.5.3 TAA's implementation

Since this is the second phase of the implementation, we take advantage of all configurations already done in TAA's environment.

In phase 1, we configured the identity feed for Tivoli Identity Manager, as described in 9.6.2, "Design considerations" on page 368. In the current phase, we implemented automatic provisioning for TAA's employees for common accounts. And, as described in Table 9-5 on page 381, an employee may be in one of three states:

- ▶ Active
- ▶ Leave of absence
- ▶ Inactive

During the first phase TAA decided that the active and inactive employee states will map to the TAAEmpStatus attribute. They also decided that employees who are on a leave of absence will remain in Tivoli Identity Manager but have an employee status of *leave of absence*. To accomplish this requirement, we again use the erPersonStatus attribute. The TAAEmpStatus is the attribute that becomes the principal point to determinate the person status within TAA.

When suspending a person, the accounts associated with this person must be suspended as well. To achieve this, the "Include accounts when suspend, restoring and deleting users" check box must be checked in the Manage Users window.

There are two different types of termination:

- ▶ Immediate
- ▶ Normal

Immediate termination refers to employees whose access to accounts must be revoked right away without any delay (such as moving on to work at a competitor, and so on). Normal termination refers to employees going through the normal process of termination where there is no requirement to immediately revoke all the departing employee's accounts. In other words, a normal termination will allow the HR feed process to run its course.

In a normal termination situation, the HR feed sets the erPersonStatus attribute to 2, which marks the person as terminated in Tivoli Identity Manager. Tivoli Identity Manager then suspends all accounts belonging to the employee.

Then, in three months' time, the user's accounts must be deleted. This is accomplished through a custom operation in the person entity. This operation will loop through all accounts that a person has and then deprovisions the account (Figure 10-8).

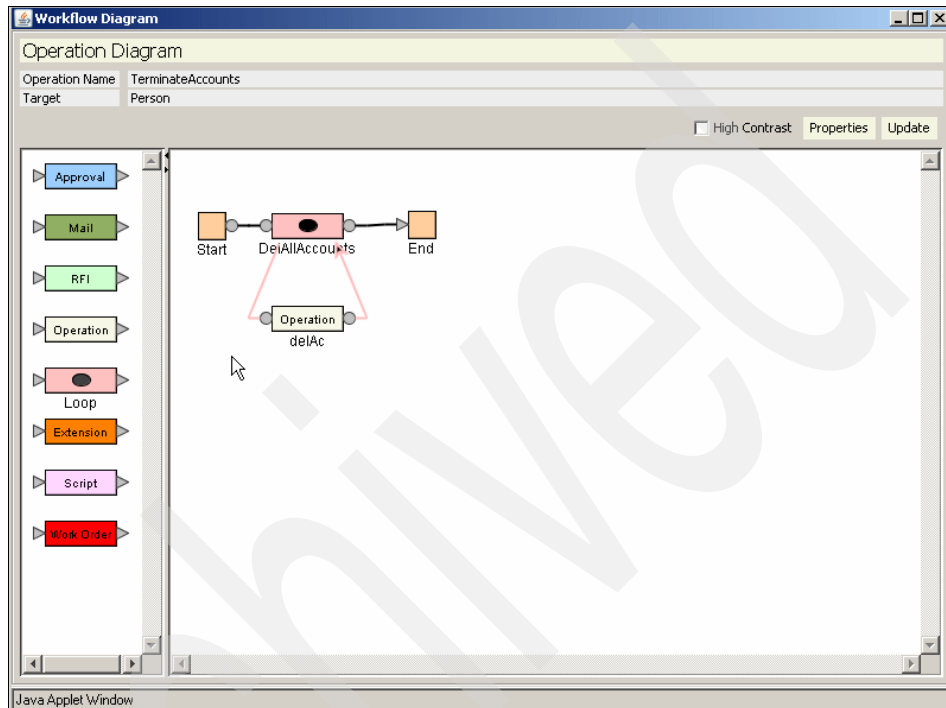


Figure 10-8 TerminateAccounts custom operation in person entity type

Table 10-10 shows the individual scripts that are tied to the different nodes in the operation workflow shown in Figure 10-8.

Table 10-10 Custom operations scripts

Node	Script
Start	accounts=Entity.get().getProperty('account'); AccountList.set(accounts);
Loop	loopcount <= AccountList.get().length
Operation	AccountList.get()[loopcount - 1]

Once this custom operation is created, a life cycle rule is required such that Tivoli Identity Manager checks for person objects with erPersonStatus of 2 every three months.

The recertification rule is set up to look for Tivoli Identity Manager person objects with `erPersonStatus` equal to 2 (that is, terminated) every quarter (Figure 10-9). When such objects are found, the `TerminateAccounts` operation is kicked off and all accounts with owners that have been terminated are deleted accordingly. This life cycle rule can also be run on an ad hoc basis.

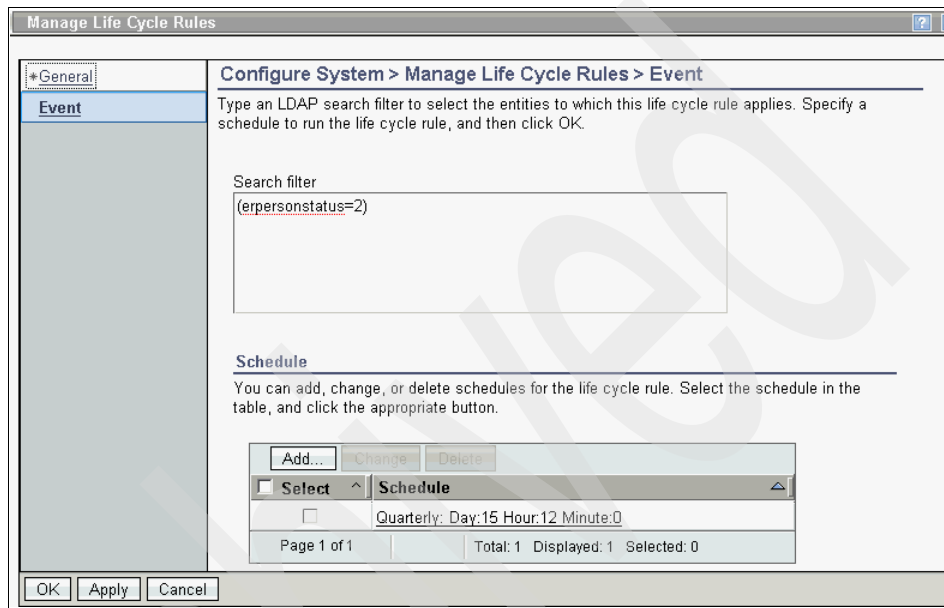


Figure 10-9 *TerminatePersonAccount* life cycle rule

To restore suspended user accounts (for example, users returning from leave of absence) you can use the default Tivoli Identity Manager process.

10.6 Reporting considerations

IBM Tivoli Manager offers predefined reporting functions for monitoring the policies configured in this chapter. Table 10-11 is a list of suggestions. For brevity, sample reports are not shown.

Table 10-11 Reports for monitoring account creation and password change activities

Purpose	Predefined Tivoli Identity Manager reports
Identify password changes.	Account operations Account operations performed by an individual
Identify suspended accounts and the associated persons and services.	Suspended accounts
Identify persons that have suspended accounts.	Suspended individuals
Identify accounts that do not comply with the password policy.	Noncompliant accounts
Identify inactive accounts that are candidates for deletion.	Dormant accounts
Identify operations performed by a Tivoli Identity Manager user.	Operation report

Technical implementation: Phase III

In this chapter we provide details on the functional requirements identified in 8.2 “Functional requirements” on page 325, that are relevant to this phase.

Table 11-1 shows the mapping of the functional requirements detailed in 8.2 “Functional requirements” on page 325, against implementation tasks detailed in 8.4 “Implementation approach” on page 335, relevant to this phase.

Table 11-1 Requirements for phase III

Functional requirement	TAA's deployment requirements
B. Password resets will be delegated to users other than the system administrators, possibly to the end users.	<ul style="list-style-type: none"> ▶ Self password reset using challenge questions ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Delegation (password resets)
C. Common values are entered automatically.	<ul style="list-style-type: none"> ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Delegation (accounts)

Functional requirement	TAA's deployment requirements
D. Manually entered values can be checked for correctness.	<ul style="list-style-type: none"> ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Delegation (accounts, compliance alerts)
E. Provide a common user interface for administration.	<ul style="list-style-type: none"> ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Change control for the Tivoli Identity Manager configuration
F. Allow delegation of approval responsibilities.	<ul style="list-style-type: none"> ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Delegation (accounts)
G. Support collaboration by multiple approvers.	<ul style="list-style-type: none"> ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Delegation (accounts)
H. Remind approvers of waiting requests.	<ul style="list-style-type: none"> ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Delegation (accounts)
I. Escalate ignored requests.	<ul style="list-style-type: none"> ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Delegation (accounts, compliance alerts)
K. Automatically add and remove accounts and access rights when a user changes job roles.	<ul style="list-style-type: none"> ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Delegation (accounts, compliance alerts)
L. Account changes made outside of the common interface are detected and checked against the security policies.	Delegation (accounts, compliance alerts)
M. Changes to security policies are checked against existing accounts.	Delegation (accounts, compliance alerts)
N. An administrator can create or change an account even if the resulting account violates the corporate security policies.	<ul style="list-style-type: none"> ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Delegation (accounts, compliance alerts)

Functional requirement	TAA's deployment requirements
O. Designated administrators will be notified when noncompliant accounts are detected.	Delegation (accounts, compliance alerts)
P. The designated administrators can decide how long the account may remain noncompliant. After this period expires the account will be automatically brought into compliance with the security policies.	<ul style="list-style-type: none"> ▶ Account management using the Tivoli Identity Manager Web user interface ▶ Delegation (accounts, compliance alerts)
V. A reporting mechanism will be available that identifies accounts that are not in compliance with the corporate security policies.	Account management using the Tivoli Identity Manager Web user interface
DD. Centralized group management	Group management for the most important platforms (Active Directory and Linux servers).

Each of the following sections can be traced back (either directly or indirectly) to the deployment requirements outlined in Table 11-1 on page 449 and detail the specifics required to achieve them. These are structured as follows:

- ▶ Information that is relevant during the design phase
- ▶ Design options and considerations
- ▶ How it was implemented at Tivoli Austin Airlines

11.1 Password challenge/response

This section discusses the requirements, design considerations, and implementation of a common password challenge/response policy for accounts on all managed services.

11.1.1 Requirements

Provide TAA personnel who forget their passwords with the ability to reset their passwords.

11.1.2 Design considerations

Some design considerations are:

- ▶ Challenge questions should be composed in a way to ask for information that is only known by that specific person.
- ▶ The challenge policy should require correct answers to multiple questions. The more diverse questions and the more questions to be answered, the stronger the policy.
- ▶ The first time that a user logs on to Tivoli Identity Manager, either through the self-service interface or the administrative console, the user should answer multiple password challenge questions. The way this is implemented differs between the two interfaces. All Tivoli Identity Manager users should therefore be educated in how to manage their challenge questions, either in the self-service interface or the administrative console.
- ▶ The password challenge policy should randomly display multiple questions that were previously selected and answered by the user.

11.1.3 TAA's implementation

From the task list in the administrative console in Tivoli Identity Manager:

1. Select the task **Set System Security**.
2. Select the **Configure Forgotten Password Settings** menu.

3. Enter the values for the challenge response rules specified in Table 11-2.

Table 11-2 Challenge response rules

Rule	Value
Enable forgotten password authentication.	Yes. (The check box contains a check mark.)
Login behavior: When the user successfully answers the questions.	Log in to system. (The radio button contains a check mark.)
Login behavior: Message suspending account for failed answers.	The account is suspended because the number of consecutive unsuccessful logon attempts is exceeded. Contact the help desk for assistance
Login behavior: Send message to e-mail address.	This should be an e-mail address to which the information about the suspended account via challenge/response will be sent.
Challenge behavior: Challenge-response behavior.	Administrator provides predefined questions.
Challenge behavior: User has a choice of predefined questions?	Random selected.
Challenge behavior: Number of questions user sets up.	5
Challenge behavior: Number of correct answers user must enter	2

See Figure 11-1 for the results.

Set System Security > Configure Forgotten Password Settings

To configure questions for users who forget their password, enable forgotten password authentication. Specify user-defined or administrator-predefined questions. If the user creates the questions, type the number of questions and answers the user creates. If the administrator predefines the questions, enter the questions, and click OK.

Enable forgotten password authentication

Login Behavior

When the user successfully answers the questions

Login to system
 Reset and e-mail password

Message suspending account for failed answers

The account is suspended because the number of consecutive failed attempts has exceeded the maximum number of failed attempts.

Send message to e-mail address

Challenge Behavior

Challenge-response behavior

Users define their own questions
 Administrator provides predefined questions

▶ [Specify Forgotten Password Question](#)

User has a choice of predefined questions?

No, answer all questions
 Yes, user selects which questions to answer
 No, answer a subset of questions that the system provides

*Number of question user sets up

5

*Number of correct answers user must enter

2

Figure 11-1 Challenge response rules

4. Expand the section **Specify Forgotten Password Question** to define the remaining challenge response questions.

- Enter the challenge questions specified in Table 11-3.

Table 11-3 Challenge questions

Rule	Value
Locale	Any locale. (Note: As the questions below are specified in English, this is the default language for the questions.)
Question (Note: This challenge list is not generally limited to five, as specified in the number of challenges/responses that a user must configure. This list is a pool of challenges ranging from number >= number of challenges/responses that a user must configure to a maximum of 100 from which the user will configure five in this case.)	What is the last name of your favorite author? What is your favorite animal? What is your favorite color? Where is your favorite place to relax? What is the first name of your favorite actor?

See Figure 11-2 for the result.

Specify Forgotten Password Question

New challenge question Locale Add

Select	Locale	Question
<input type="checkbox"/>	Any locale	What is the last name of your favorite author?
<input type="checkbox"/>	Any locale	What is your favorite animal?
<input type="checkbox"/>	Any locale	What is your favorite color?
<input type="checkbox"/>	Any locale	Where is your favorite place to relax?
<input type="checkbox"/>	Any locale	What is the first name of your favorite actor?

Page 1 of 1 Total: 5 Displayed: 5 Selected: 0

Figure 11-2 Challenge questions

- Click the **OK** button to save this information and you will see a window stating that you have successfully changed the forgotten password configuration and a summary of the changes.

Note: TAA has also implemented Spanish language support for all of its corporate applications. We just showed the definition for the challenge/response feature using English as the default language, but TAA also defined them for the Spanish language.

11.2 Account management using the Web user interface

A major benefit of an Tivoli Identity Manager implementation is to provide people within the organization with the ability to manage their accounts centrally through a common interface. This section details the relevant steps involved.

11.2.1 Requirements

It is a requirement of TAA to have the ability to perform centralized account management by administrators and users alike. This also must be centrally auditable and managed. Depending on which tasks users are going to perform, allowing users access to either the Tivoli Identity Manager self-service or administrative console Web interfaces meets this requirement.

11.2.2 Design considerations

Allowing access for a person to Tivoli Identity Manager is a matter of provisioning the person with an Tivoli Identity Manager account and, if access is required to the administrative console, adding the account to a Tivoli Identity Manager group that has the appropriate view associated with it permitting access to the admin console. Note that this is different from their person record within Tivoli Identity Manager. The existence of a person within Tivoli Identity Manager does not imply the ability for him to access the Tivoli Identity Manager's Web interfaces. The person's existence within Tivoli Identity Manager merely illustrates the fact that he is known to Tivoli Identity Manager and can have accounts managed for him. This includes the actual account to the Tivoli Identity Manager application, that is, the Tivoli Identity Manager account.

Care must be taken in determining the set of people within the organization for which a Tivoli Identity Manager account is to be provisioned. People who do not need to use Tivoli Identity Manager do not necessarily need to have an Tivoli Identity Manager account. The determining factor is usually dependent on the stage of the implementation and whether the requirements at the time mandate the fact that people need access to any of the two Tivoli Identity Manager Web interfaces. This can range from having people manage their own information, reset their passwords, manage their accounts, perform administration, or participate in workflow actions such as approvals. All these actions require that a person is provisioned with an Tivoli Identity Manager account.

Basic provisioning of an Tivoli Identity Manager account gives access to the Tivoli Identity Manager self-service interface. This provides users with the ability to view their own person entry and account details and accesses, and reset their passwords and request new accounts and accesses. Varying levels of access to these can be enabled or disabled through the configuration of the relevant access control items, groups, and views in Tivoli Identity Manager. Figure 11-3 shows the standard page that a user sees when logging into Tivoli Identity Manager.

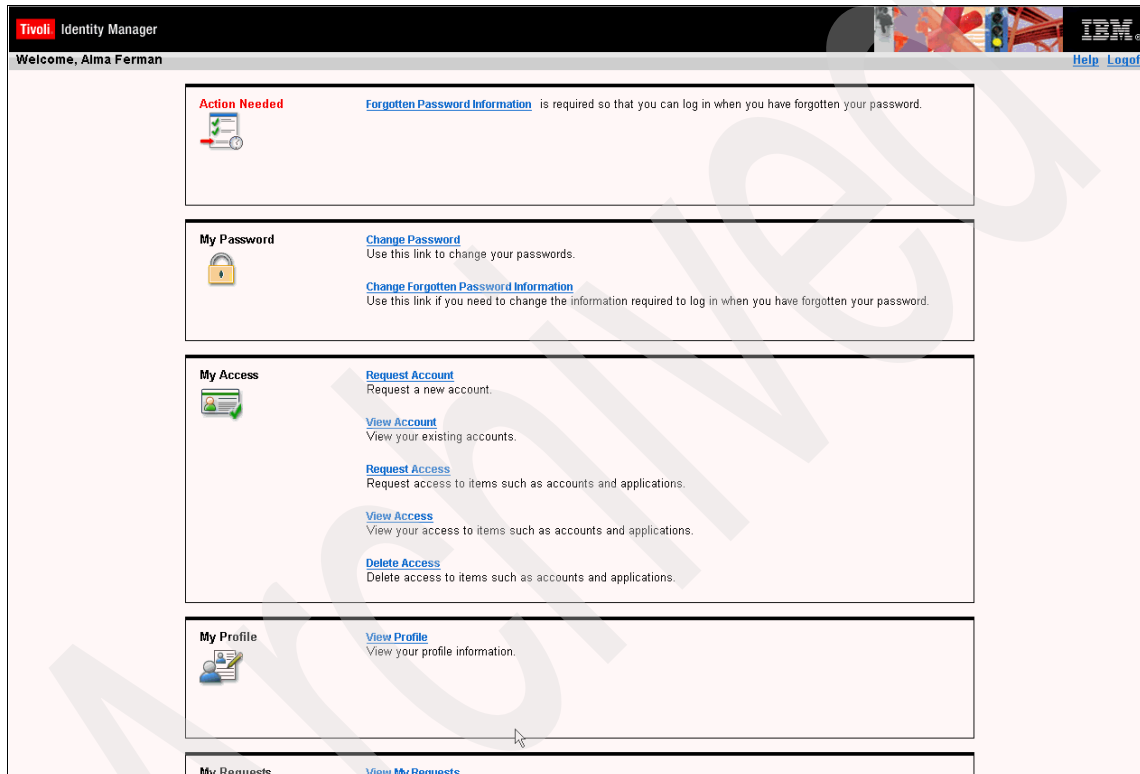


Figure 11-3 Tivoli Identity Manager self-service interface for a standard user

Note: The self-service interface provides several functions for users. It can be accessed byway of a browser, but there are some special situations when the self-service interface is not accessible to users. When a user cannot access its own desktop because its account is either blocked or a user forgets the password, there is no way to load the interface. For this reason, TAA has considered installing the Tivoli Identity Manager Desktop Password Reset Assistant (DPRA) on every desktop. See Appendix B, “Windows desktop password reset and unlock” on page 631 for a detailed explanation.

The level of access that a person has into the Tivoli Identity Manager Web interfaces is based on the views and ACI settings that are attached to the Tivoli Identity Manager groups, which may be affected by the Tivoli Identity Manager groups that they are a member of (that is, the Tivoli Identity Manager groups that their Tivoli Identity Manager accounts are members of), and also whether they are defined as a managers for other people. As per best practice, these should be driven by provisioning policies. Standard access into Tivoli Identity Manager is also driven by provisioning policies, and, by default, Tivoli Identity Manager installs with a standard provisioning policy that allows all people to have Tivoli Identity Manager accounts, as shown in Figure 11-4, Figure 11-5 on page 459, Figure 11-6 on page 459, Figure 11-7 on page 460, and Figure 11-8 on page 460.

Manage Policies > Manage Provisioning Policies > General

Specify information for the policy, the business unit to which the policy applies, and the scope of the policy within the organization. When you are done specifying information on each of the tabs, click Preview to review your changes, or click Save as Draft if you want to save your changes and finish this definition at a later time. Click Submit to save your changes now. Click Cancel to exit without saving your changes.

*Policy name
Default provisioning policy for ITIM

Caption
ITIM account policy

Make policy available to services in
 This business unit and its subunits
 This business unit only

Description
Allow everyone to be provisioned for an ITIM account.

Policy status
 Enable
 Disable

*Priority (integer greater than 0)
10000000

Keywords

*Business unit
Tivoli Austin Airlines

Submit Preview... Save as Draft Cancel Search...

Figure 11-4 Default Tivoli Identity Manager provisioning policy General tab

<ul style="list-style-type: none"> *General *Members *Entitlements 	<p>Manage Policies > Manage Provisioning Policies > Members</p> <p>Members are the set of users that are granted entitlements through a policy. Specify which members are granted the entitlements that are defined in this policy by selecting all users in the organization, individual roles, or all users who are not defined in other policies. If you choose to select the roles, you can only select existing roles.</p> <p>*Member Type</p> <p><input checked="" type="radio"/> All users in the organization</p> <p><input type="radio"/> All other users who are not granted to the entitlement(s) defined by this provisioning policy via other policies</p> <p><input type="radio"/> Roles specified below</p>
<input type="button" value="Submit"/> <input type="button" value="Preview..."/> <input type="button" value="Save as Draft"/> <input type="button" value="Cancel"/>	

Figure 11-5 Default Tivoli Identity Manager provisioning policy Members tab

<ul style="list-style-type: none"> *General *Members *Entitlements 	<p>Manage Policies > Manage Provisioning Policies > Entitlements</p> <p>Specify the entitlements that are associated with this policy. You can select and then change and delete the attributes of an entitlement.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="4" style="text-align: center;"> <input type="button" value="Create"/> <input type="button" value="Change"/> <input type="button" value="Delete"/> <input type="button" value="Parameters"/> </td> </tr> <tr> <th style="text-align: left;">Select</th> <th style="text-align: left;">Name</th> <th style="text-align: left;">Target type</th> <th style="text-align: left;">Provision Option</th> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td>ITIM Service</td> <td>Specific Service</td> <td>Automatic</td> </tr> <tr> <td colspan="2" style="text-align: center;">Page 1 of 1</td> <td colspan="2" style="text-align: center;">Total: 1 Displayed: 1 Selected: 0</td> </tr> </table>	<input type="button" value="Create"/> <input type="button" value="Change"/> <input type="button" value="Delete"/> <input type="button" value="Parameters"/>				Select	Name	Target type	Provision Option	<input type="checkbox"/>	ITIM Service	Specific Service	Automatic	Page 1 of 1		Total: 1 Displayed: 1 Selected: 0	
<input type="button" value="Create"/> <input type="button" value="Change"/> <input type="button" value="Delete"/> <input type="button" value="Parameters"/>																	
Select	Name	Target type	Provision Option														
<input type="checkbox"/>	ITIM Service	Specific Service	Automatic														
Page 1 of 1		Total: 1 Displayed: 1 Selected: 0															
<input type="button" value="Submit"/> <input type="button" value="Preview..."/> <input type="button" value="Save as Draft"/> <input type="button" value="Cancel"/>																	

Figure 11-6 Default Tivoli Identity Manager provisioning policy Entitlements tab

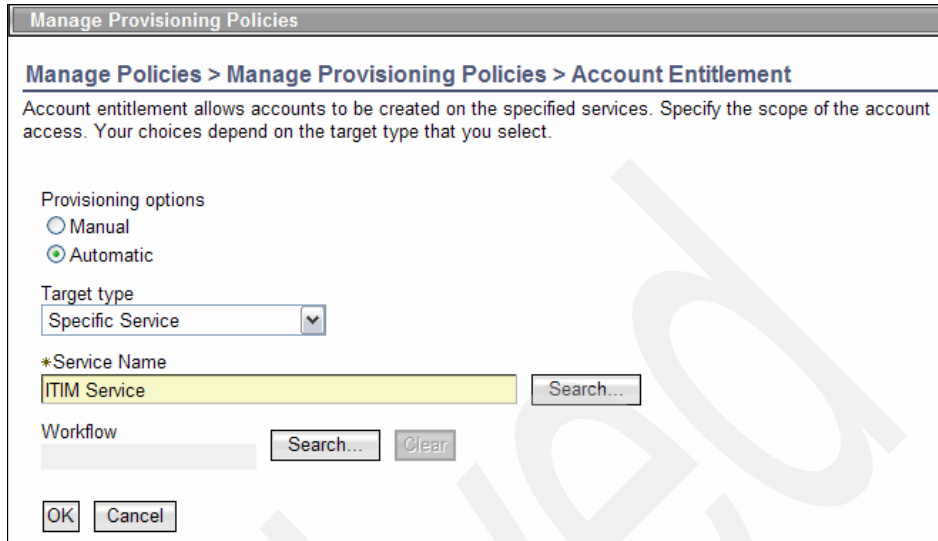


Figure 11-7 Default Tivoli Identity Manager provisioning account entitlement

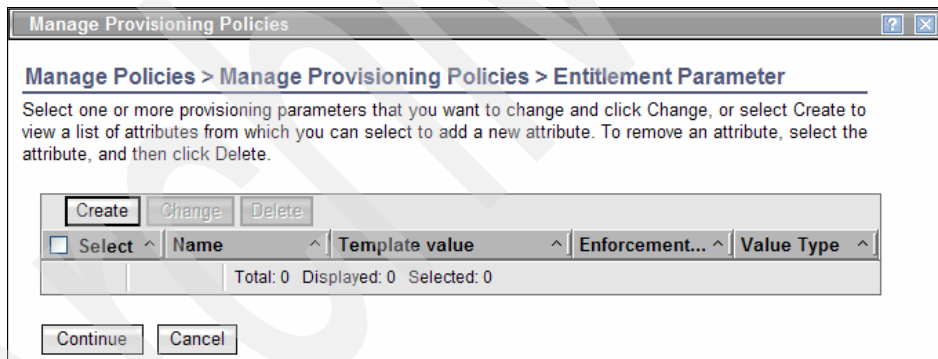


Figure 11-8 Default Tivoli Identity Manager provisioning policy entitlement parameters

The decision must be made to determine whether this default provisioning policy can be left as is or whether it should be modified, disabled, or deleted. Leaving it as is allows for the possibility of people having Tivoli Identity Manager accounts.

Requirements may determine that the policy be modified or be combined with another policy. For example, another provisioning policy may cover the entitlements for the standard Tivoli Identity Manager account holder and hence may negate the need for this standard provisioning policy. In certain cases, it may be required to disable or delete the policy if there is a requirement to have restrictive controls over the ability to be provisioned with an Tivoli Identity Manager account. That is, if there is no requirement to have Tivoli Identity Manager accounts provisioned to all people defined within Tivoli Identity Manager, it may be required that the provisioning policy be disabled, deleted, or modified to reflect this.

11.2.3 TAA's implementation

This phase requires that general access to the Tivoli Identity Manager Web interface is enabled for all users. Varying levels of access are required for different types of users, and these are covered in 11.3 "Delegation" on page 461. The only requirement that TAA has allows all people whose accounts are managed by Tivoli Identity Manager to have access to the Web interface to perform tasks. Standard users should only be able to reset their passwords and view their information.

TAA's requirement can be satisfied by using the standard provisioning policy.

11.3 Delegation

In a heterogeneous environment, there are usually many systems and processes to deal with and to manage, and no single person will typically know all the business rules required to administrate them. As a result, there is usually a business procedural model implemented to allow delegation of administration and business functions to allow for better controls on the environment.

11.3.1 Requirements

TAA requires that password resets, account creation, and account modification no longer be handled by a central administrator as was the case in phases I and II. These should be delegated to people's managers and to the users in the case of password resets. The business process tasks also must be modelled, managed, actioned, and audited within Tivoli Identity Manager. This includes approvals of account creations and modifications, and the notification and actioning of non-compliant accounts. Also, service owners will have the ability to manage groups (create, modify membership, and delete) for Microsoft Active Directory resources.

11.3.2 Design considerations

Before discussing the various areas within Tivoli Identity Manager where delegation of administration may be implemented, there must be consideration given to the delegation approach in general.

Delegation approach

There must be some thought put into the usage of admin domains and access control items (ACIs). The decision to use admin domains is driven in part by the structure of the organization tree and also the business requirements. ACIs are normally used in some form to control access to operations and data within Tivoli Identity Manager, although the use of ACIs can range from very simple to extremely complex.

Access control items

These control the finer-grained access controls within the Tivoli Identity Manager organization tree. This is independent of whether it is within standard elements within the tree such as locations or organization units or within administrative domains. An ACI controls access to data and operations on the object that it is protecting. This differs depending on the object. For example, search, modify, suspend, add, restore, transfer, and remove operations on a person object and search, modify, add, remove, suspend, and restore operations on an account object. Read and write access to specific data attributes for the relevant object can also be specified in the ACI. The level of access to grant to specific users within Tivoli Identity Manager for delegation of management are different for each deployment and for each requirement because of the differing business rules that must be represented and enforced. The aim of this section is not to discuss the specifics of the ACIs, but to identify the need to analyze the access control requirements for each type of user and implement the relevant ACIs to enforce the controls. For more information about ACIs, refer to Chapter 4, “Detailed component design” on page 101, or the IBM Tivoli Identity Manager Information Center Version 5.1.

Admin domains

Admin domains offer a convenient way to reuse a single set of ACIs in different portions of the organization tree with the ACIs applying to different sets of users. An admin domain exists within the Tivoli Identity Manager organization tree as a branch of the tree in the same way that an organization unit or location exists as branches of the organization tree. One or more Identity Manager users may be designated as the administrator of a domain. The difference between an admin domain and the other container types is in the way that ACIs are enforced with respect to the administrators of admin domains.

Every ACI has a set of *principals*. These are the Identity Manager users who are granted or denied some accesses by the ACI. An ACI's principals can be defined by one or more Tivoli Identity Manager groups, or by dynamically calculated relationships. One of these dynamically calculated relationships is domain administrator. Any ACI granting a right to domain administrators grants that right to all domain administrators, but only within their own domains. This assumes that the admin domains themselves are in the scope of the ACI.

Figure 11-9 shows how a single ACI granting a right to domain administrators grants that right to different domain administrators in different locations. This could be accomplished without admin domains, but doing so would require replacing the single ACI with separate ACIs granting rights to Tom in his subtree and to Ann in her subtree.

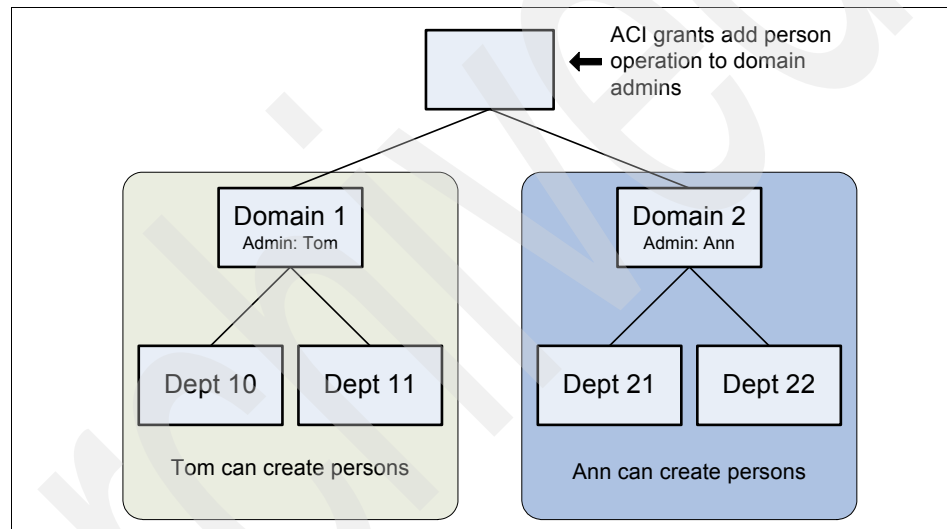


Figure 11-9 A single ACI grant rights to different domain administrators

Identity Manager has a default set of ACIs that grant rights to domain administrators. These ACIs make domain administrators almost the equals of Identity Manager administrators. The only rights not given to domain administrators are the functions available to Identity Manager administrators in the Web interface's Configuration tab, and the ability to view any users' requests and to-do list items. You can delete the default domain admin ACIs if you would like to define more restrictive rights for your domain administrators.

For more information about admin domains, refer to the IBM Tivoli Identity Manager Information Center Version 5.1.

Delegation areas

There are various distinct areas to consider when dealing with delegation in Tivoli Identity Manager:

- ▶ People and account management: Includes actions such as account creations, modifications, and password resets
- ▶ Business processes: Includes actual auditable business tasks required to ensure that the necessary business processes are put in place to assist with meeting the audit and business compliance controls of the organization such as relevant approvals tasks and account compliance enforcement
- ▶ Service and policy management: Includes the management of the service definitions in Tivoli Identity Manager for the managed resources and also the management of the provisioning, identity, and password policies
- ▶ Access control definition: Includes the administration of the access controls defined by access control items

Delegating administration purely through the use of administrative domains automatically gives the domain administrator the full set of privileges to control these four areas within their domain. This may serve the purposes of delegation in some cases. In others, there may need to be finer-grained access controls implemented within the administrative domains to cater for a more distinct separation of duties. For example, a person may only be allowed to perform account management operations but not be able to change provisioning policies. This holds true irrespective of the decision to use an administrative domain. That is, ACIs are required for finer-grained control within Tivoli Identity Manager in general.

People and account management

ACIs control people's details and operations on their records and accounts in Tivoli Identity Manager. An ACI where the target object is *erPersonItem* applies to people entity objects. ACIs where the target object is *erAccountItem* apply to accounts in general. These object types are the generic representations of person objects and account objects. ACIs allow for explicit application to specific person object types (in cases where there are custom object types for person objects) and specific account types (for each account type managed by Tivoli Identity Manager).

Figure 11-10 shows the default ACI defined within Tivoli Identity Manager that allows all operations to be performed on person objects by their managers (supervisors), domain administrators, sponsors, and help desk users.

***General**

Operations

Permissions

Membership

Set System Security > Change Access Control Item > General

To change the name for the access control item, type in a new name. Review the read-only information for the protection category and type, and then click OK.

*Name
Default ACI for Person: Grant All to Supervisor/Domain Admin/Sponsor/Help Desk Group

Protection Category
Person

Type
erPersonItem

Apply object protection on this business unit
Tivoli Austin Airlines

and...
 all of its sub units

Apply protection to...
 All objects in the selected category or class
 A subset of objects that satisfy the filter criteria

OK Apply Cancel

Figure 11-10 Default ACI for person object

Figure 11-11 shows the same ACI but with the specification that all attributes can be viewed and modified by a person's manager, domain administrators, sponsors, and help desk users.

Set System Security > Change Access Control Item > Operations

Select the permission that defines how users will be allowed to perform each operation, and then click OK.

Select all permissions
Grant

Operation	Permission
Add	Grant
Change Password	Grant
Modify	Grant
Remove	Grant
Restore	Grant
Search	Grant
Suspend	Grant
Transfer	Grant

Page 1 of 1 Total: 8 Displayed: 8

OK Apply Cancel

Figure 11-11 Default ACI for person object (continued)

Figure 11-12 shows the ACI permissions granted on the person attributes.

Set System Security > Change Access Control Item > Permissions

Select the permission that defines how users will be allowed to perform each attribute, and then click OK.

Select all read Select all write
Grant Grant

Attribute	Read	Write
Aliases	Grant	Grant
E-mail address	Grant	Grant
Full name	Grant	Grant
Last Name	Grant	Grant
Last Operation	Grant	Grant
Locale	Grant	Grant
Organizational roles	Grant	Grant
Others	Grant	Grant
Shared secret	Grant	Grant
Supervisor	Grant	Grant

Page 1 of 1 Total: 10 Displayed: 10

OK Apply Cancel

Figure 11-12 Default ACI for person object (continued)

Figure 11-13 shows who is a member of this ACI (that is, the person's manager, domain administrators, sponsors, and members of the help desk group).

Set System Security > Change Access Control Item > Membership

Select the focus for **person** entity access that is governed by this access control item.

- All users in the system
- The profile owner
- The manager of the profile owner
- The sponsor of the business partner organization in which the person resides
- The administrator of the domain in which the person resides
- Users who are members of these groups

<input type="checkbox"/> Select	Group Na...	Description
<input type="checkbox"/>	Help Desk Assistant	Default Help Desk Assistant Group for Tivoli Austin Airlines

Page 1 of 1 Total: 1 Displayed: 1 Selected: 0

OK Apply Cancel

Figure 11-13 Default ACI for person object (continued)

Figure 11-14 shows a different ACI. This ACI is also created by default in an Tivoli Identity Manager installation and allows for people to change their own passwords. To be able to view their accounts to change passwords, they are also granted search permissions.

The screenshot displays the 'Set System Security > Change Access Control Item > General' configuration window. On the left, a navigation pane shows 'General' selected, with other options like 'Operations', 'Permissions', and 'Membership'. The main area contains the following fields and options:

- *Name:** A text field containing 'Default ACI for Account: Grant Search, Add, Change Password, and All groupMember operations to Self'.
- Protection Category:** A dropdown menu set to 'Account'.
- Type:** A dropdown menu set to 'erAccountItem'.
- Apply object protection on this business unit:** A dropdown menu set to 'Tivoli Austin Airlines'.
- and...:** A checkbox labeled 'all of its sub units' which is checked.
- Apply protection to...:** Two radio button options: 'All objects in the selected category or class' (selected) and 'A subset of objects that satisfy the filter criteria'.
- Filter criteria:** A large, empty text area for defining filter criteria.

At the bottom of the window, there are three buttons: 'OK', 'Apply', and 'Cancel'.

Figure 11-14 Default ACI for account

Figure 11-15 shows the part of this same ACI that allows for the update of the password attribute.

Select all permissions
-- Select --

Operation	Permission
Add	Grant
Add Group Member	Grant
Modify	Grant
Remove	None
Remove Group Member	Grant
Restore	None
Search	Grant
Suspend	None
View Group Member	Grant

Page 1 of 1 Total: 9 Displayed: 9

OK Apply Cancel

Figure 11-15 Default ACI for account (continued)

Figure 11-16 depicts the Permissions window.

Select all read Select all write
-- Select -- -- Select --

Attribute	Read	Write
Access last certified date	None	None
Access recertification last action	None	None
Last Operation	None	None
Last recertification action	None	None
Object Type	None	None
Others	None	None
Owner	None	None
Password	Grant	Grant
Service	None	None
User ID	Grant	None

Page 1 of 1 Total: 10 Displayed: 10

OK Apply Cancel

Figure 11-16 Default ACI for account (continued)

Figure 11-17 shows the Membership window.

*General

Operations

Permissions

Membership

Set System Security > Change Access Control Item > Membership

Select the focus for account entity access that is governed by this access control item.

- All users in the system
- The account owner
- The manager of the account owner
- The owner of the service that the account resides on
- The owner of any access defined on the service that the account resides on
- The sponsor of the business partner organization in which the account resides
- The administrator of the domain in which the account resides
- Users who are members of these groups

Add Remove

Select	Group Name	Description

Total: 0 Displayed: 0 Selected: 0

OK Apply Cancel

Figure 11-17 Default ACI for account (continued)

These are not the only standard ACIs in an Tivoli Identity Manager installation. Analysis of the business requirements must be performed to determine whether the standard set of ACIs meets the requirements and whether they must be modified or deleted, or whether new ACIs must be created.

Business processes

Delegation of business processes is not based purely on ACIs. These usually involve actions such as (but not limited to) approvals and are defined within workflows, either as account workflows, access workflows, or operation workflows. As a result, ACIs only play their part in allowing the person actioning the process to view the relevant data when they are logged in. In certain cases, they may not need the ACI access to the data, as some workflow items show the details of the change as part of the description text. This must be a design decision. In a simple scenario, there does not usually need to be much ACI configuration performed.

Workflow participants can generally be distinct individuals or persons performing a role. This is not the same as performing the analysis required to control provisioning based on a role-based access control model. Roles in this context are simply a way to identify a set of people who should be actioning a particular step in a business process. Note that the use of roles here is not mutually exclusive to roles used for role-based access control in provisioning. That is, there is nothing stopping the workflow from using a role used to control provisioning.

Service and policy management

As with the person and account access controls, these are controlled by ACIs. Service ACIs are similar to person and account ACIs in that they can target a generic concept of a service (that is, all services) or specific service types, such as all Windows Active Directory services.

Delegating administration of these items is typically given to business process administrators with domain knowledge of Tivoli Identity Manager. The separation of duties between a person performing administration on people and their accounts and someone performing the business and system controls governing the policies that enforce the behavior of operations on people and accounts is sometimes desired. That is, organizations do not always want a person who can manage the services and policies defined within Tivoli Identity Manager also to have the ability to manage people and accounts, and vice versa. There may also be audit and compliance requirements that enforce this separation of duties.

Figure 11-18 shows the general section of the default ACI that allows for domain administrators to perform all operations on a service.

The screenshot displays the 'Set System Security > Change Access Control Item > General' dialog box. On the left, a navigation pane shows 'General' selected, with other options like 'Operations', 'Permissions', and 'Membership'. The main area contains the following fields and options:

- Name:** A dropdown menu showing 'Default ACI for Service: Grant All to Domain Admin'.
- Protection Category:** A text field containing 'Service'.
- Type:** A text field containing 'erServiceItem'.
- Apply object protection on this business unit:** A text field containing 'Tivoli Austin Airlines'.
- and...:** A checkbox labeled 'all of its sub units' which is checked.
- Apply protection to...:** Two radio buttons: 'All objects in the selected category or class' (selected) and 'A subset of objects that satisfy the filter criteria'.
- Buttons:** 'OK', 'Apply', and 'Cancel' at the bottom.

Figure 11-18 Default ACI for services

Figure 11-19 shows the same ACI but with the specification that all attributes on a service can be viewed and modified by the domain administrator.

Set System Security > Change Access Control Item > Operations

Select the permission that defines how users will be allowed to perform each operation, and then click OK.

Select all permissions
-- Select --

Operation	Permission
Add	Grant
Modify	Grant
Recertification Override	None
Reconcile	Grant
Remove	Grant
Search	Grant

Page 1 of 1 Total: 6 Displayed: 6

OK Apply Cancel

Figure 11-19 Default ACI for services (continued)

Figure 11-20 illustrates the attribute permissions granted for domain admins on a service.

Set System Security > Change Access Control Item > Permissions

Select the permission that defines how users will be allowed to perform each attribute, and then click OK.

Select all read Select all write
Grant Grant

Attribute	Read	Write
Description	Grant	Grant
Others	Grant	Grant
Owner	Grant	Grant
Service name	Grant	Grant
Service prerequisite	Grant	Grant

Page 1 of 1 Total: 5 Displayed: 5

OK Apply Cancel

Figure 11-20 Default ACI for services (continued)

Figure 11-21 shows the membership configuration for this ACI.

***General**

Operations

Permissions

Membership

Set System Security > Change Access Control Item > Membership

Select the focus for **service** entity access that is governed by this access control item.

- All users in the system
- The supervisor of the business unit in which the service resides
- The owner of the service
- The owner of any access defined on the service
- The sponsor of the business partner organization in which the service resides
- The administrator of the domain in which the service resides
- Users who are members of these groups

Add Remove

Select	Group Name	Description

Total: 0 Displayed: 0 Selected: 0

OK Apply Cancel

Figure 11-21 Default ACI for services (continued)

Figure 11-22, Figure 11-23 on page 475, Figure 11-24 on page 475, and Figure 11-25 on page 476 illustrate the same concepts for a provisioning policy.

***General**

Set System Security > Change Access Control Item > General

To change the name for the access control item, type in a new name. Review the read-only information for the protection category and type, and then click OK.

+Name

Default ACI for Provisioning Policy: Grant All to Domain Admin/Service Owner Group

Protection Category
Provisioning Policy

Apply object protection on this business unit
Tivoli Austin Airlines

and...

all of its sub units

Apply protection to...

All objects in the selected category or class

A subset of objects that satisfy the filter criteria

OK Apply Cancel

Figure 11-22 Default ACI for provisioning policy

***General**

Operations

Permissions

Membership

Set System Security > Change Access Control Item > Operations

Select the permission that defines how users will be allowed to perform each operation, and then click OK.

Select all permissions
Grant

Operation	Permission
Add	Grant
Modify	Grant
Remove	Grant
Search	Grant

Page 1 of 1 Total: 4 Displayed: 4

OK Apply Cancel

Figure 11-23 Default ACI for provisioning policy (continued)

***General**

Operations

Permissions

Membership

Set System Security > Change Access Control Item > Permissions

Select the permission that defines how users will be allowed to perform each attribute, and then click OK.

Select all read Select all write
Grant Grant

Attribute	Read	Write
Description	Grant	Grant
Enabled	Grant	Grant
Entitlements	Grant	Grant
Keywords	Grant	Grant
Label	Grant	Grant
Original policy DN	Grant	Grant
Policy Item Name	Grant	Grant
Policy membership	Grant	Grant
Policy Target	Grant	Grant
Priority	Grant	Grant
Service Resolution Scope	Grant	Grant

Page 1 of 1 Total: 11 Displayed: 11

OK Apply Cancel

Figure 11-24 Default ACI for provisioning policy (continued)

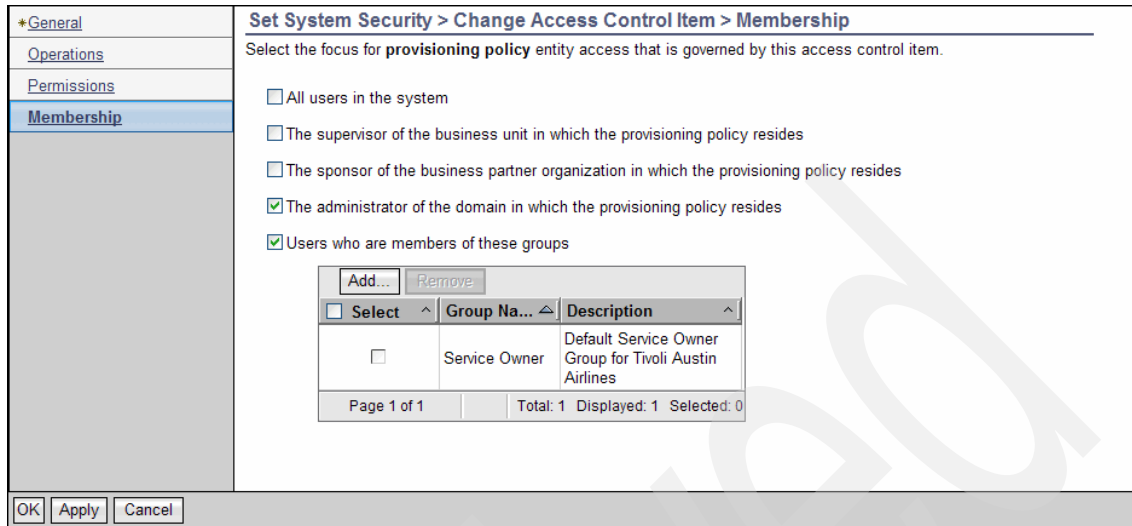


Figure 11-25 Default ACI for provisioning policy (continued)

It is common to have organizations not see the need to delegate service and policy administration to users. They specify that these continue to be handled by Tivoli Identity Manager administrators. This is the more simple approach and is the one adopted by most organizations unless there is a business or regulatory need to enforce the distinct separation of duties to not allow administrators of policies and services the same type of control over people and their accounts. This is because it is usually more crucial that administrators who manage people and accounts not have the access to modify the services and policies, rather than enforce that administrators of services and policies not have the access to administrate people and accounts.

Access control definition

Tivoli Identity Manager provides the ability to delegate the administration of the ACIs defined in each part of the organization tree. This is specified via the authorization owner tab accompanying the list of ACIs for each part of the organization tree, as shown in Figure 11-26. Members of the groups specified in the list of authorization owners can administrate the ACIs within the relevant part of the organization tree.

Set System Security > Manage Access Control Items > ACI Owners

To add or remove an owner of access control items in Tivoli Austin Airlines business unit, select an owner in the table, and then click the appropriate button.

Select	Group Name	Business unit
<input type="checkbox"/>	ACI admin	Tivoli Austin Airlines

Page 1 of 1 Total: 1 Displayed: 1 Selected: 0

OK Cancel

Figure 11-26 ACI Owners menu

Organization tree design and ACI placement

ACIs grant or deny rights to perform a set of actions on a given type of object if that object is at the same position in the organization tree as the ACI or, optionally, in the subtree below the ACI's position. This means that your delegation model will often have a great influence on the design of your organization tree.

For example, it is common to place all services and provisioning policies at the top level of the tree and have the Identity Manager administrators manage them all. But let us say that you want to delegate the maintenance of Lotus Notes provisioning policies to a group of Notes administrators. There is no way that you can do this if the Notes provisioning policies are at the top level of the tree. Granting someone the right to create or modify a provisioning policy gives them that right regardless of what service type is the target of the policy. So granting your Notes administrators these rights on provisioning policies would also allow them to create and modify provisioning policies for Active Directory and RACF.

However, provisioning policies can only have target services that are at or below the organization tree location of the provisioning policy. You can accomplish your goal by creating a Notes Provisioning container in the organization tree, and creating ACIs on that container that grant your Notes administrators the rights to create and modify Lotus Notes services and provisioning policies. By restricting the possible set of target services to just the Notes services that they can create, you have effectively restricted them to maintaining just the Notes provisioning policies.

Group management

Tivoli Identity Manager provides additional security administration through group management capabilities. There is a list of adapters that support this capability. For more information about group management, refer to IBM Tivoli Identity Manager Information Center Version 5.1.

11.3.3 TAA's implementation

TAA has decided to continue to administrate ACIs, services, and policies using Tivoli Identity Manager administrators. However, it still requires that account creation and modification be delegated to people's managers (supervisors) and password resets be allowed to be done by the person or his manager.

Workshops with the business and system owners at TAA to refine the initial requirement to have business processes modeled in Tivoli Identity Manager and audited resulted in the following:

- ▶ Each account create/modify request must be approved by a system administrator of the specific managed resource instance. For example, all Linux account creations and modifications must be approved by the Linux administrator for that service.
- ▶ Compliance alerts are to be generated on all Linux accounts and sent to the relevant regional audit team.
- ▶ Compliance alerts are to be generated on all Windows Active Directory accounts and sent to the relevant regional audit team.
- ▶ Escalation periods for inaction are to be set to one day.
- ▶ All escalations are to be sent to the Tivoli Identity Manager system administrator.

To support the requirements for this phase, there has also been the need to refine the provisioning policies. For example, provisioning policies are the basis of compliance alerts. To support this, the provisioning policies must be more restrictive than previous phases.

Policies

Note that this phase does not use role-based access control enforcement of provisioning policies. That is planned for phase IV. The policies, however, must be more restrictive, specifically for the Linux and Windows Active Directory accounts, to ensure that people only have accounts that are relevant for their location. For example, a person in the central region should not have a Linux account for the server in the east region. In order to reduce and simplify the number of provisioning policies that must be created, TAA used a service selection policy for Linux and for Windows Active Directory accounts.

To illustrate the benefit of using service selection policies in this case, consider one of the design alternatives.

Design alternative

This alternative involves the use of dynamic roles to indicate membership based on a person's location in the organization tree and have a provisioning policy for each location. Each provisioning policy targets the specific service instances and applies the provisioning policy based on the relevant dynamic role. Table 11-4 details the provisioning policies required for the east region alone.

Table 11-4 Provisioning policies for East Region

Provisioning policy	Applies to	Services allowed
East region provisioning policy	People located in the east region organization unit in the organization tree—implemented using a dynamic role based on erparent attribute.	<ul style="list-style-type: none"> ▶ East region Linux ▶ New York CSC Active Directory
Detroit CSC provisioning policy	People located in the Detroit CSC organization unit in the organization tree—implemented using a dynamic role based on erparent attribute.	Detroit CSC Active Directory
New York CSC provisioning policy	People located in the New York CSC organization unit in the organization tree—implemented using a dynamic role based on erparent attribute.	New York CSC Active Directory
Raleigh CSC provisioning policy	People located in the Raleigh CSC organization unit in the organization tree—implemented using a dynamic role based on erparent attribute.	Raleigh CSC Active Directory

Consider the fact that Table 11-4 only covers the east region and that similar policies must apply to the central and west regions, and it becomes apparent that making this design decision requires many provisioning policies. In the case of TAA, this means 14 provisioning policies (10 for the CSCs, one for the corporate HQ Linux, and three for the regional centers). This does not take into account the additional provisioning policy required to cover the fact that all users are allowed to have Tivoli Access Manager, Lotus Notes, and RACF accounts.

The solution

The use of service selection policies has allowed TAA to meet the requirements with the use of a single provisioning policy because the selection of the specific services to provision is scripted into the service selection policy.

Linux service selection policy

The service selection policy for Linux accounts in the TAA environment is:

```
function selectService(){
    var services = ServiceSearch.searchForClosestToPerson(subject);
    if(services != null && services.length > 0){

if(subject.getProperty("parent")[0].name==services[0].getProperty("parent")[0].
name){
    return services[0];
    }
    }
}
return selectService();
```

This uses the `ServiceSearch.searchForClosestToPerson` function to find the Linux service defined in the closest part of the organization tree to the relevant person. There is also additional scripting that only allows people to have Linux accounts if the service resides in the same part of the organization tree as the person. This ensures that only the people in the regional centers can have Linux accounts. Without it, people in the CSCs are allowed to have Linux accounts.

Windows Active Directory service selection policy

The service selection policy for Windows Active Directory accounts in the TAA environment is:

```
function selectService(){
    var services = ServiceSearch.searchForClosestToPerson(subject);
    if(services != null && services.length > 0){
        return services[0];
    }
    else{
        if(subject.getProperty("parent")[0].name=="Central Region"){
            services = ServiceSearch.searchByFilter("(erservicename=Austin CSC Active
Directory)", 2);
            return services[0];
        }
        else if(subject.getProperty("parent")[0].name=="East Region"){
            services = ServiceSearch.searchByFilter("(erservicename=New York CSC
Active Directory)", 2);
            return services[0];
        }
        else if(subject.getProperty("parent")[0].name=="West Region"){
```

```

        services = ServiceSearch.searchByFilter("(erservicename=San Francisco CSC
Active Directory)", 2);
        return services[0];
    }
}
}
return selectService();

```

This also uses the `ServiceSearch.searchForClosestToPerson` function to find the Windows Active Directory service defined in the closest part of the organization tree to the relevant person. Note that there is additional scripting put in place to ensure that the regional administrators are also allowed to have Windows Active Directory accounts. They are assigned accounts on the relevant domain for their region, that is, Austin for the central region, New York for the east region, and San Francisco for the west region. This is required because there is no Windows Active Directory service defined at the regional level and using the `ServiceSearch.searchForClosestToPerson` function does produce a result, as all the services are defined at lower levels in the organization tree and are not covered by the search scope of the function. That is, the function does not return a service for people in the regional centers because the function does not search for services in lower levels of the organization tree.

Workflow

TAA uses an account workflow to enforce the business process requirements. Figure 11-27 shows the details of the General tab within the workflow details.

The screenshot shows a web-based configuration interface for a workflow. The breadcrumb navigation is "Design Workflows > Manage Account Request Workflows > General". The main content area contains the following fields:

- *Name:** A text input field containing "ServiceAdminApproval".
- Description:** A text area with a scroll bar, currently empty.
- *Business unit:** A text input field containing "Tivoli Austin Airlines" and a "Search..." button to its right.
- *Service type:** A dropdown menu with "All" selected.

At the bottom of the form, there are three buttons: "OK", "Apply", and "Cancel".

Figure 11-27 TAA account approval workflow: General details

Figure 11-28 illustrates the flow of the approval process in the workflow.

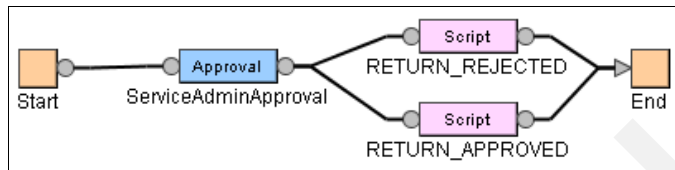


Figure 11-28 TAA workflow approval: Logical flow

As per the business requirement, the workflow has a single approval by the service administrator. Figure 11-29 shows the details of the General tab within the approval node in the workflow. The main aspect to note is the custom participant.

ID	Type	Relevant Data ID
entity	Account	entity
service	Service	service
owner	Person	owner

Figure 11-29 TAA workflow approval node details

To avoid having to define the exact same workflow multiple times with only a single difference (the participant—a different one for each service instance), TAA decided to use a naming convention combined with a custom script to determine the relevant participant to use for account create/modify approvals. TAA has

enforced the naming convention that each service name must have a corresponding role defined to handle approvals for that service. The name of this role must be the name of the service followed by a space and then the word *Approvers*. Hence, for each service defined in phase I, there is a corresponding approval role. The script defined within the approval node's custom participant that implements the logic to handle the routing of approvals is:

```
var serviceName = service.get().name;
var results = (new RoleSearch()).searchByName(serviceName + ' Approvers');
if(results.length!=1)
    return new Participant(ParticipantType.SYSTEM_ADMIN);
else
    return new Participant(ParticipantType.ROLE, results[0].dn);
```

Notice that any requests that cannot be resolved to a relevant role are routed to the Tivoli Identity Manager administrator.

Provisioning policy

It has been determined that a single provisioning policy leveraging the workflow shown in Figure 11-28 on page 482 and the service selection policies for Windows Active Directory and Linux are sufficient to meet the requirements of phase III. The information specified in the General tab of the provisioning policy is shown in Figure 11-30.

Manage Policies > Manage Provisioning Policies > General

Specify information for the policy, the business unit to which the policy applies, and the scope of the policy within the organization. When you are done specifying information on each of the tabs, click **Preview** to review your changes, or click **Save as Draft** if you want to save your changes and finish this definition at a later time. Click **Submit** to save your changes now. Click **Cancel** to exit without saving your changes.

*Policy name
Global Accounts Provisioning Policy

Caption

Make policy available to services in
 This business unit and its subunits
 This business unit only

Description

Policy status
 Enable
 Disable

*Priority (integer greater than 0)
10000

Keywords

*Business unit
Tivoli Austin Airlines

Figure 11-30 TAA phase III global accounts provisioning policy: General tab

As the concept of role-based access control is not being implemented in this phase, the provisioning policy applies to all users, as shown in Figure 11-31.

Manage Policies > Manage Provisioning Policies > Members

Members are the set of users that are granted entitlements through a policy. Specify which members are granted the entitlements that are defined in this policy by selecting all users in the organization, individual roles, or all users who are not defined in other policies. If you choose to select the roles, you can only select existing roles.

*Member Type

- All users in the organization
- All other users who are not granted to the entitlement(s) defined by this provisioning policy via other policies
- Roles specified below

Submit Preview... Save as Draft Cancel

Figure 11-31 TAA phase III global accounts provisioning policy: Membership tab

The provisioning policy must still apply to all services. That is, it must allow people to have Tivoli Access Manager accounts, Lotus Notes accounts, RACF accounts, Linux accounts, and Windows Active Directory accounts.

As shown in Figure 11-32, the Tivoli Access Manager and Lotus Notes e-mail entitlements are now set to automatic as per the requirement in phase II. The provisioning policy also must be configured to enforce the workflow shown in Figure 11-28 on page 482 to all services covered by the provisioning policy.

Manage Policies > Manage Provisioning Policies > Entitlements

Specify the entitlements that are associated with this policy. You can select and then change and delete the attributes of an entitlement.

<input type="checkbox"/> Select	Name	Target type	Provision Option
<input type="checkbox"/>	TAA Access Manager Service	Specific Service	Automatic
<input type="checkbox"/>	Lotus Notes Email	Specific Service	Automatic
<input type="checkbox"/>	POSIX Linux profile	Service Selection Policy	Manual
<input type="checkbox"/>	Active Directory Profile	Service Selection Policy	Manual
<input type="checkbox"/>	Central IT Data Center RACF	Specific Service	Manual

Page 1 of 1 Total: 5 Displayed: 5 Selected: 0

Submit Preview... Save as Draft Cancel

Figure 11-32 TAA phase III global accounts provisioning policy: Entitlements tab

Of the accounts allowed for TAA users, Linux and Windows Active Directory accounts must be more tightly controlled to be selective of the service instance. This is enforced via the use of the service selection policies defined and is shown in Figure 11-33 on page 487 and Figure 11-34 on page 487. Tivoli Access Manager, Lotus Notes, and RACF accounts still continue to target the relevant service instance, as there exists only a single service instance for each of these managed resources. Note that RACF accounts should only be allowed for Central IT Data Center developers, but this is not being enforced in this phase.

Manage Policies > Manage Provisioning Policies > Account Entitlement

Account entitlement allows accounts to be created on the specified services. Specify the scope of the account access. Your choices depend on the target type that you select.

Provisioning options

Manual
 Automatic

Target type
Service Selection Policy ▼

Service type
Active Directory Profile ▼

Governing service selection policy name
Active Directory service selection policy

Workflow
ServiceAdminApproval

Figure 11-33 TAA phase III global accounts provisioning policy: AD entitlement detail

Manage Policies > Manage Provisioning Policies > Account Entitlement

Account entitlement allows accounts to be created on the specified services. Specify the scope of the account access. Your choices depend on the target type that you select.

Provisioning options

Manual
 Automatic

Target type
Service Selection Policy ▼

Service type
POSIX Linux profile ▼

Governing service selection policy name
Linux service selection policy

Workflow
ServiceAdminApproval

Figure 11-34 TAA phase III global accounts provisioning policy: Linux entitlement detail

Account creation/modification

The requirement to have managers create and modify their employees' details and accounts is mostly met by the default Tivoli Identity Manager groups, views, and ACIs. Some examples of the ACIs that allow for supervisors to manage their employees person entities in Tivoli Identity Manager are shown in "People and account management" on page 464. The *Default ACI for Account: Grant All to Supervisor/Domain Admin/Sponsor/Service Owner/Access Owner*, as shown in Figure 11-35, grants full access to a person's accounts to their supervisor, domain administrator, sponsor, service owner, or access owner.

The screenshot displays the 'Set System Security > Change Access Control Item > General' configuration window. On the left is a navigation pane with tabs for *General, Operations, Permissions, and Membership. The main area contains the following fields and options:

- Name:** A dropdown menu showing 'Default ACI for Account: Grant All to Supervisor/Domain Admin/Sponsor/Service Owner/Access Owner'.
- Protection Category:** A text field containing 'Account'.
- Type:** A text field containing 'erAccountItem'.
- Apply object protection on this business unit:** A text field containing 'Tivoli Austin Airlines'.
- and...:** A checkbox labeled 'all of its sub units' which is checked.
- Apply protection to...:** Two radio button options: 'All objects in the selected category or class' (selected) and 'A subset of objects that satisfy the filter criteria'.
- Filter Criteria:** A large, empty text area below the radio buttons.

At the bottom of the window are three buttons: 'OK', 'Apply', and 'Cancel'.

Figure 11-35 TAA default ACI

Figure 11-36 shows the attribute permission details for the ACI shown in Figure 11-35 on page 488. Here TAA modified the default ACI grant all on account object to supervisor/domain admin/sponsor/service owner/access owner attribute permission details.

Set System Security > Change Access Control Item > Permissions

Select the permission that defines how users will be allowed to perform each attribute, and then click OK.

Select all read: Grant
Select all write: Grant

Attribute	Read	Write
Access last certified date	Grant	Grant
Access recertification last action	Grant	Grant
Last Operation	Grant	Grant
Last recertification action	Grant	Grant
Object Type	Grant	Grant
Others	Grant	Grant
Owner	Grant	Grant
Password	Grant	Grant
Service	Grant	Grant
User ID	Grant	Grant

Page 1 of 1 Total: 10 Displayed: 10

OK Apply Cancel

Figure 11-36 TAA modified default ACI

There are two additional tasks that must be done to enable a person's manager to be able to perform all the relevant tasks required to manage his person and account details:

- ▶ Allow managers access to the administrative console of Tivoli Identity Manager to manage their employees' accounts.
- ▶ Modify the default manager view in Tivoli Identity Manager to allow managers to change employees' person and account details.

Access to the administrative console

During the identity feed of TAA employee data into Tivoli Identity Manager, the manager was defined for each person entity. Tivoli Identity Manager can use this information to automatically populate the default Manager group with the Tivoli Identity Manager accounts of these users. Adding the managers into the default manager group automatically grants access to the administrative console through the default manager view and ACIs created during installation. To enable this feature, expand the *Set System Security* navigation menu in the administrative console and select the **Set Security Properties** option. Check the box under the **Group Settings** called *Automatically populate identity manager groups*, as shown in Figure 11-37.

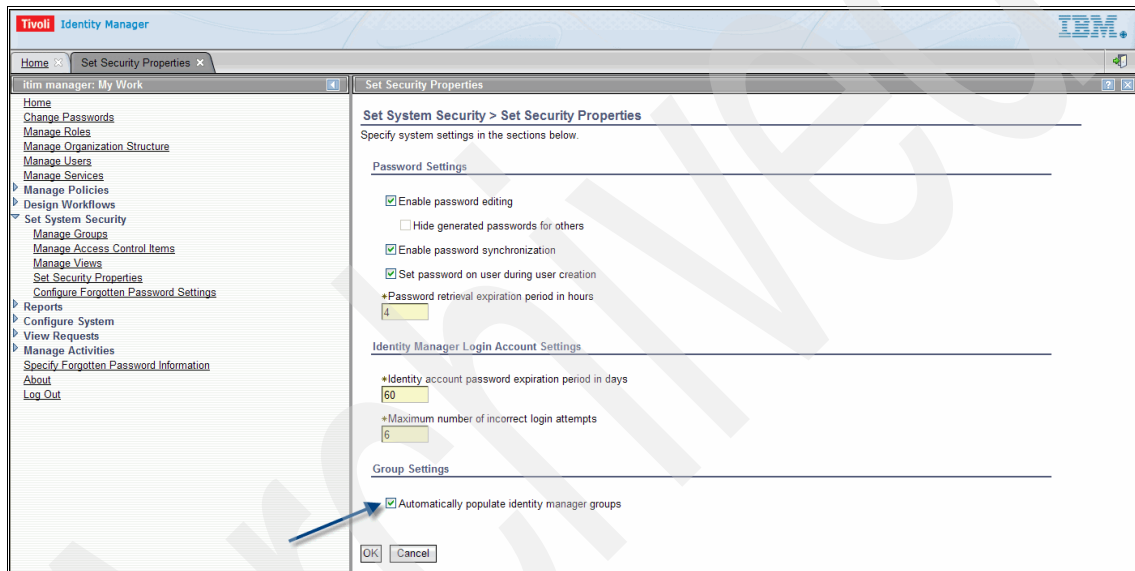


Figure 11-37 Automatically populate identity manager groups

The automatic action is enabled or disabled immediately after pressing the **OK** button to submit the request. You do not need to restart Tivoli Identity Manager. Figure 11-38 illustrates the start page after logging in to Tivoli Identity Manager using an account that is a member of the default manager group.

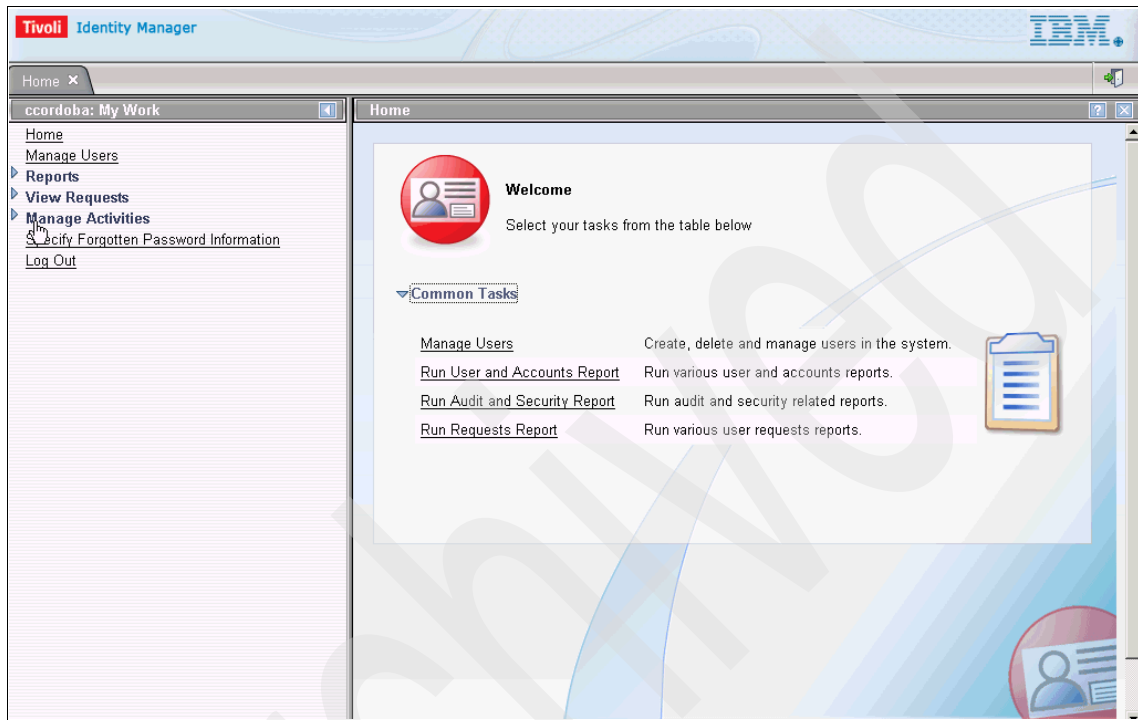


Figure 11-38 The standard manager view in the administrative console

Modify the default manager view

The default manager view in Tivoli Identity Manager grants managers access to the administrative console where they are allowed to perform certain tasks. The default ACIs described earlier in “People and account management” on page 464 granted managers all operations and attribute permissions to manage their employees person and account details. Using the default Manager view puts some restrictions to this.

The default Manager view allows managers to perform the following tasks in the administrative console:

- ▶ Suspend and restore users.
- ▶ Change user passwords.
- ▶ Delegate activities.
- ▶ Request, delete, suspend, and restore accounts.

It does not allow managers to create new users, change existing users, or change existing accounts and account passwords belonging to their employees.

In order to fulfill the TAA requirement that all managers should be able to manage their employees' person and account details, the default Manager view must be changed. As the creation process of users is taken care of by the identity feed from the TAA HR system, managers will not be allowed the task to create users. They will also not take care of transferring or deleting users with HR-type activities or that are handled by automatic life cycle rules within Tivoli Identity Manager. To configure the correct set of tasks for TAA managers, select **Manage Views** under the Set System Security menu. Then search for the name *manager view* to find and select the correct view. Go to the **Configure View** tab, as shown in Figure 11-39 on page 493, and ensure that the following items are checked under the admin console tasks:

- ▶ Manage Users: Change User
- ▶ Manage Users: Suspend User
- ▶ Manage Users: Restore User
- ▶ Manage Users: Change Passwords
- ▶ Manage Users: Delegate Activities
- ▶ Manage User Accounts: Request an Account
- ▶ Manage User Accounts: Change an Account
- ▶ Manage User Accounts: Delete an Account
- ▶ Manage User Accounts: Suspend Accounts
- ▶ Manage User Accounts: Restore Accounts

All other tasks within the view are left with their default settings.

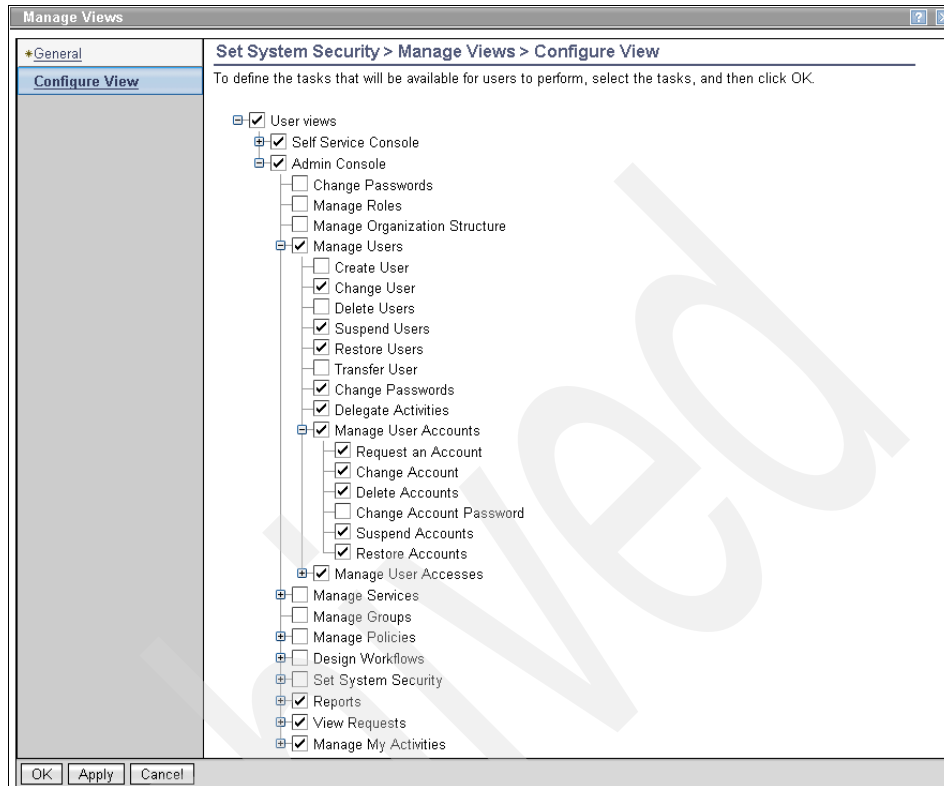


Figure 11-39 Configuring the manager view

Password resets

By virtue of being able to perform actions on their employee details and accounts, managers can also reset their passwords. There is an additional requirement here that users be able to reset their own passwords through the Tivoli Identity Manager self-service interface. This is enabled by default, as the ACI in Figure 11-14 on page 468 shows. Because of the default ACI, the requirement is therefore also met. There is no need to add users to any groups as was the case in “Account creation/modification” on page 488, where managers needed additional access to the administrative console. This is because the set of windows in the Tivoli Identity Manager self-service interface allowed for a standard user includes the ability to reset a user’s password. This functionality becomes available by default when a user is provisioned with an Tivoli Identity Manager account, which was not implemented until this phase.

Compliance alerts

Linux and Windows Active Directory accounts must generate compliance alerts when they are no longer within policy. Within the confines of the implementation of phase III, this can occur if a person transfers between regions or CSCs. For example, a person transferring from the west regional center to the central regional center will continue to have her west regional Linux account. This is no longer compliant to the service selection policy, which mandates that they should have a central region Linux account because of their location.

Figure 11-40 shows the west region Linux service policy enforcement definition for compliance alerts. Alerts are sent to people who are members of the west region security and compliance role and escalated to the Tivoli Identity Manager system administrator after one day. The maximum deferral limit is set to 90 days. This is the longest deferral that the security and compliance administrators can grant to a non-compliant account. For example, they may only grant a deferral period of 5 days. The non-compliance will be corrected by Tivoli Identity Manager at the end of the deferral period granted or at the expiration of the escalation period.

***General**
E-mail

Manage Services > Configure Policy Enforcement Behavior > General

To create an alert for West region Linux service, select the participants and time intervals. Also, specify the process types for which an alert is generated, and then click Submit.

*Alert name
Compliance alert for West region

Send compliance alert to
Organizational Role

Organizational role name
West Region Security and Compliance

Number of days to wait before escalating compliance alert
1

Escalate compliance alert to
System Administrator

Number of days after which the system will take corrective action
90

Process Types

Select the process types for which an alert is generated. If no process type is selected, the system automatically corrects a non-compliant account for that process type. The correction can modify or delete the account.

<input type="checkbox"/> Generate Alert	Process Type
<input checked="" type="checkbox"/>	Reconciliation
<input checked="" type="checkbox"/>	Policy change
<input checked="" type="checkbox"/>	Modification of account by user
<input checked="" type="checkbox"/>	Modification of account owner's data

Page 1 of 1 Total: 4 Displayed: 4 Selected: 4

Figure 11-40 West region Linux service policy enforcement compliance alert details

Notice that the “Modification of account owners’ data” check box is enabled. This is not enabled by default and specifies that Tivoli Identity Manager should generate compliance alerts for identity-change-related operations. Transferring between business units is one such example of an identity change. Leaving this unchecked causes non-compliant accounts to be de-provisioned when transferring between business units, as they do not fall within the rules for generating compliance alerts.

The east region Linux and central region Linux services have corresponding service policy enforcement definitions with compliance alerts being sent to members of the *east region security and compliance* and *central region security and compliance* roles, respectively.

Windows Active Directory Services defined within the CSCs also have corresponding service policy enforcement definitions with compliance alerts being sent to members of the:

- ▶ West region security and compliance role for the Los Angeles, San Francisco, and Seattle CSCs
- ▶ Central region security and compliance role for the Austin, Denver, Mexico City, and St. Louis CSCs
- ▶ East region security and compliance role for the Detroit, New York, and Raleigh CSCs

For more details refer to the section “Configuring” in the IBM Tivoli Identity Manager Information Center Version 5.1.

Group management

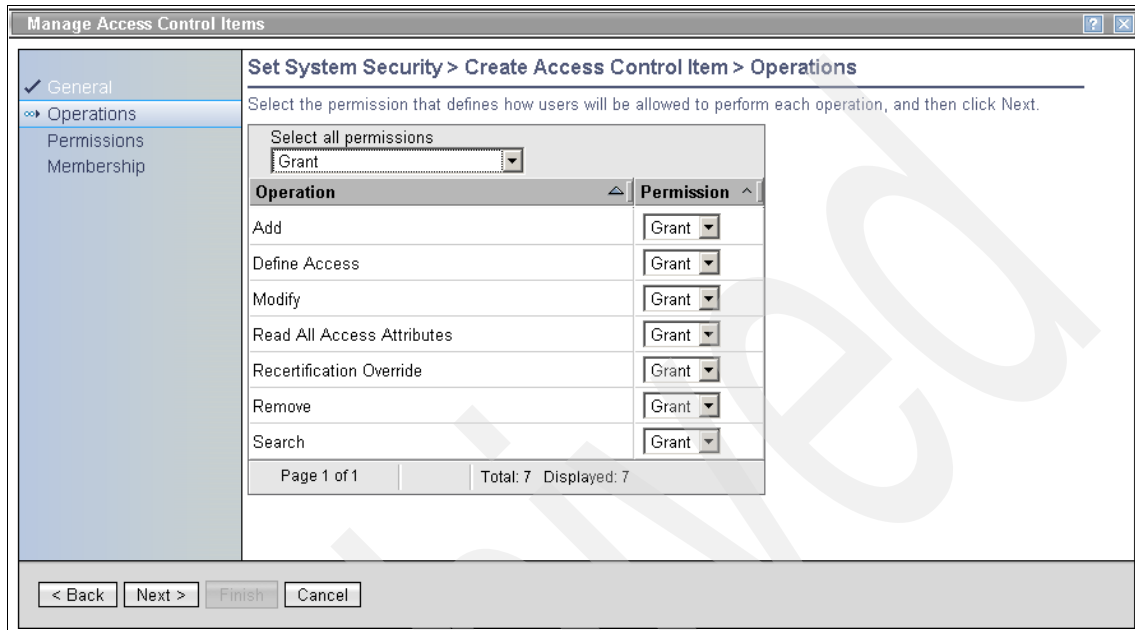
TAA’s administrators are responsible for managing groups over all main platforms and applications, such as Microsoft Active Directory and Linux Servers. In the design process, it has been established that service owners (persons that are responsible for specific resources or services) will have the ability to manage groups within the services that they own. Before group management activities can be carried out byway of Tivoli Identity Manager, you have to assign an owner to the resources. Once a service owner has been defined, it is necessary to configure an Access Control Item (ACI) that allows the service owners to perform activities within the services they own. In the following screen captures, we depict the process and values configured in TAA’s implementation.

On the main menu, go to **Set System Security** → **Manage Access Control Items** and click **Create**. The first information window needed for the ACI creation is displayed. As shown in Figure 11-41 it is very important to configure the Protection Category selection as Service Group. Click **Next** to continue.

The screenshot shows a window titled "Manage Access Control Items" with a sub-header "Set System Security > Create Access Control Item > General". The window is divided into a left sidebar and a main content area. The sidebar contains a tree view with "General" selected, and other options: "Operations", "Permissions", and "Membership". The main content area has a title bar "Set System Security > Create Access Control Item > General" and a description: "Type the name for the access control item. Also, specify a protection category and additional protection information, and then click Next." Below this are several form fields: a text field for "*Name" containing "ACI.For.Group.Management"; a dropdown for "Protection Category" set to "Service Group"; a dropdown for "Type" set to "All types"; a section for "Apply object protection on this business unit" with a search box containing "Tivoli Austin Airlines" and a "Search..." button; a checkbox "all of its sub units" which is checked; and a section "Apply protection to..." with two radio buttons: "All objects in the selected category or class" (selected) and "A subset of objects that satisfy the filter criteria". At the bottom of the window are four buttons: "< Back", "Next >", "Finish", and "Cancel".

Figure 11-41 General values for creating an ACI

In order for the service owners to manage groups, all operations must be granted to them. Grant all permissions in the Operations tab, as shown in Figure 11-42. Click **Next** to continue.



The screenshot shows a window titled "Manage Access Control Items" with a breadcrumb path: "Set System Security > Create Access Control Item > Operations". The left sidebar has "Operations" selected. The main area contains a "Select all permissions" dropdown set to "Grant" and a table of operations with their respective permissions.

Operation	Permission
Add	Grant
Define Access	Grant
Modify	Grant
Read All Access Attributes	Grant
Recertification Override	Grant
Remove	Grant
Search	Grant

Page 1 of 1 | Total: 7 | Displayed: 7

< Back | Next > | Finish | Cancel

Figure 11-42 Operation permissions

This takes you to the Permissions tab, where you need to grant all the permissions for the attributes, as shown in Figure 11-43. Click **Next** to continue.

Manage Access Control Items

Set System Security > Create Access Control Item > Permissions

Select the permission that defines how users will be allowed to perform each attribute, and then click Next.

Select all read: Grant | Select all write: Grant

Attribute	Read	Write
Group description	Grant	Grant
Group ID	Grant	Grant
Group name	Grant	Grant
Others	Grant	Grant

Page 1 of 1 | Total: 4 Displayed: 4

< Back | Next > | Finish | Cancel

Figure 11-43 Attributes permissions

On the final Membership tab, you need to define the ACI membership only for service owners, as shown in Figure 11-44 on page 499.

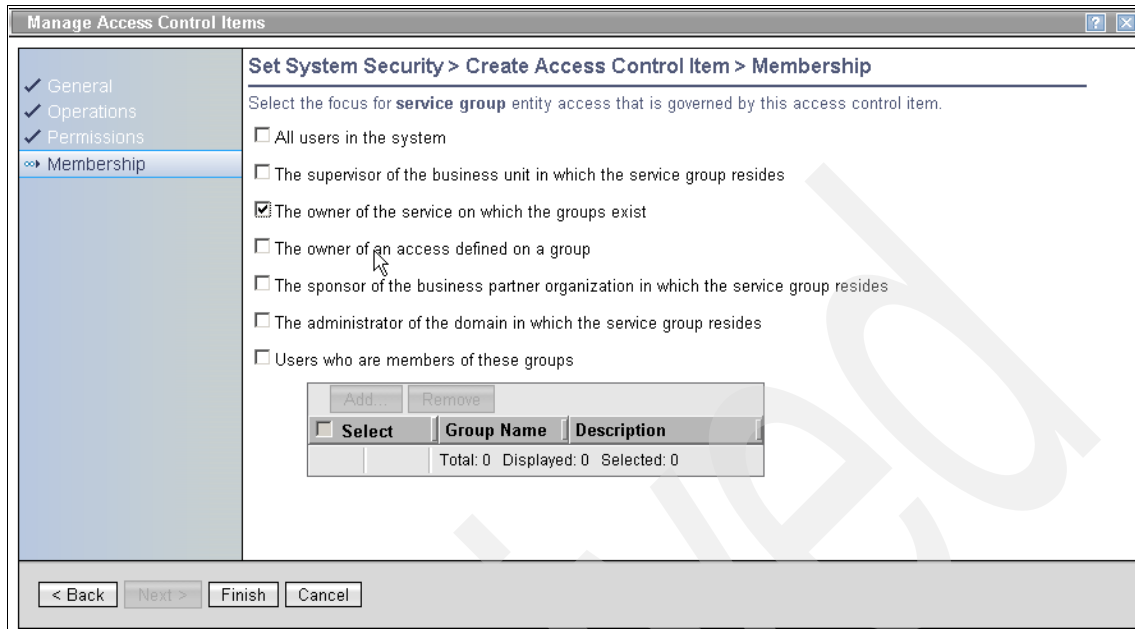


Figure 11-44 ACI membership

Now service owners have the ability to create, change, and delete groups on the target resource as long as the Tivoli Identity Manager 5.1 adapter is installed. Refer to the IBM Tivoli Identity Manager Information Center Version 5.1 for detailed information about the group management functionality. The latest information about the supported adapters can be found at the Tivoli Identity Manager support Web site at:

<http://www.ibm.com/support/search.wss?rs=644&tc=SSTFWV&atrn=Keywords&atriv=itimagentinv>

There is no need to modify any of the service owner's views in Tivoli Identity Manager. Once a service owner is logged in to the system, they can perform activities for managing groups. Figure 11-45 shows the default view for a typical service owner.

The screenshot displays the Tivoli Identity Manager web interface. On the left is a navigation menu for user 'aferman: My Work', listing options like Home, Manage Users, Manage Services, Manage Groups, Manage Policies, Design Workflows, Reports, View Requests, and Manage Activities. The main content area is titled 'Home' and features a 'Welcome' message with a 'Select your tasks from the table below' instruction. Below this is a 'Service Connection Status' section with a summary bar showing 0 Failed, 0 unknown, and 1 Success. A table below the summary shows one service: MS.AD.AustinCSC, which is a Microsoft Active Directory Service for users located at Austin CSC, with a last status date of 9/24/09 11:10:17. At the bottom of the main area is a 'Common Tasks' section with links and descriptions for tasks such as 'Manage Users', 'Manage Services', 'Run User and Accounts Report', 'Run Custom Report', 'Run Services Report', 'Run Audit and Security Report', 'Run Requests Report', and 'View Pending Requests by Service'. A clipboard icon is visible on the right side of the 'Common Tasks' section.

Status	Service Name	Service description	Last Status Date
Success	MS.AD.AustinCSC	Microsoft Active Directory Service for users located at Austin CSC	9/24/09 11:10:17

Figure 11-45 Default view for a service owner

At this point the service owner can create a new group on a managed service. Let us take a closer look at this process:

1. As the service owner, go to the **Manage Groups** → **Create Group** window. In the General Information tab, shown in Figure 11-46, you need to specify a name for your group, FlightOperationMembers. (Since this group will be created on an Active Directory system, you can choose the type of the group and all the characteristics that a group have on the Active Directory system. We are using Security Group by selecting it in the Group Type drop-down menu, and we leave the default options for the other fields. When you are finished, click **Next**.

The screenshot shows a Windows-style dialog box titled "Manage Groups". The breadcrumb path is "Manage Groups > Create Group > General Information". The main content area contains a text box for "Group unique name" with the value "FlightOperationMembers". Below it are fields for "Common Name", "Container", "Group Type" (set to "Security"), "Group Scope" (set to "Global"), and "Member of". A "Description" text box is at the bottom. Navigation buttons at the bottom include "<< Back", "Next >", "Finish", and "Cancel".

Figure 11-46 Service owner ability to create a group on an assigned service

2. In the Access Information window, you can define an access for this group. Accesses are manual permissions that can be requested for Tivoli Identity Manager users in both the administrative console and the self-service interface. We are not defining an access for this specific group. When you are finished, click **Next**.

Manage Groups > Create Group > Access Information

Select the Define an Access check box to activate the access fields. Specify the access information, such as name, type, description, and owner. Additionally, you can choose to enable access requests by users and specify whether or not the access will appear in the Common Access list. If the Define an Access check box is subsequently unchecked, the information in the fields will be cleared when the operation is completed.

Define an Access

Access status

Enable Access

Enable Common Access

Disable Access

Access name

Access type

Application

Access description

Access owner

Search Clear

Approval workflow

No Approval Required

Notify users when access is provisioned and available for use

Notify users when access is de-provisioned

< Back Next > Finish Cancel

Figure 11-47 Define access for the new group

3. Finally, in the Group Membership tab, you can specify which users will be members of this new group. When you are done, click **Finish** to create the group on the managed target.

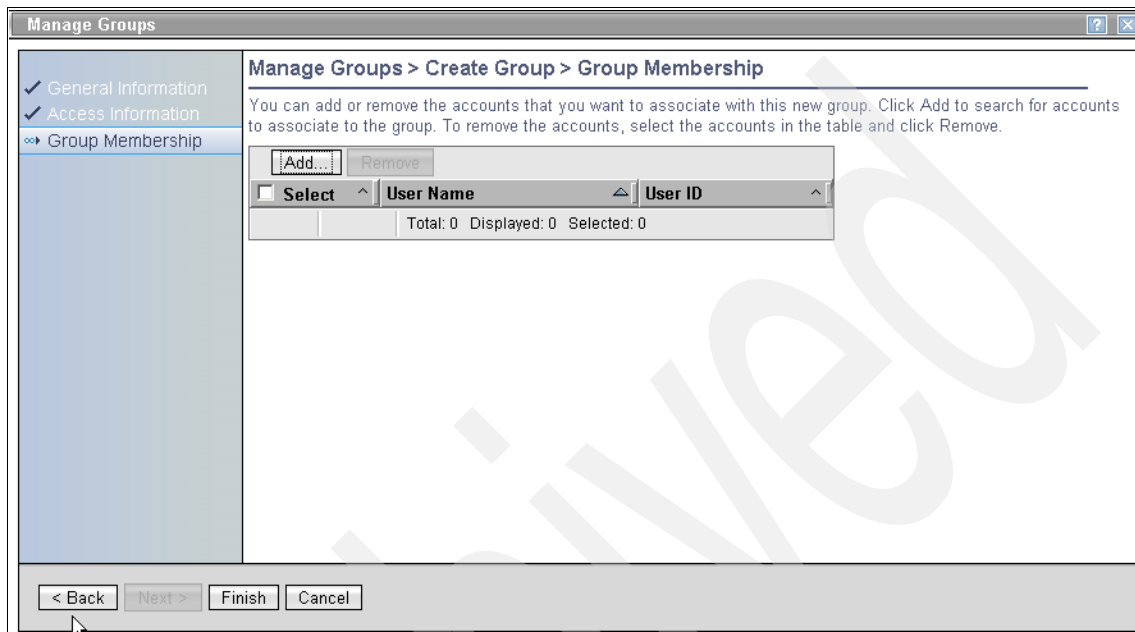


Figure 11-48 Define membership for the new group

11.4 Tivoli Identity Manager change control

The second phase of the technical implementation yielded an Identity Manager installation that is actively managing accounts in TAA's production environment. The third phase of the implementation requires the definition of new policies and functions in Identity Manager. These new features must be developed and tested without impacting the production environment that was created in phase II.

11.4.1 Requirements

TAA has no functional requirements for change control. These are non-functional requirements stemming from their IT best practices.

TAA has stated the following requirements:

- ▶ No changes will be made to the configuration of the production Identity Manager environment without the approval of the Identity Manager Change Control Board (IMCCB).

- ▶ All changes proposed for the production environment must have first been tested in an environment that approximates the production environment.
- ▶ Test environments must offer some level of isolation that protects the production environment from accidental changes resulting from actions in the test environment.

11.4.2 Design considerations

The design and implementation of new functions in Identity Manager can be approached as a software development project requiring separate development, testing, and production environments. Identity Manager provides import/export tools that can be used to promote changes to the Identity Manager configuration between these environments.

There are several issues surrounding how the different environments should be created, maintained, and used. These issues should be given much thought before you finalize your change control procedures. Some of the issues to consider are:

- ▶ How many environments are needed?
- ▶ How closely should each environment mirror the production environment?
- ▶ How will each environment be maintained?
- ▶ How will the environments be isolated from each other?

The following sections discuss these issues in detail.

How many environments are needed

Once you have placed an Identity Manager installation into production use, there should always be at least two environments:

- ▶ One for production
- ▶ One for development and testing of changes

But this is often insufficient. One of the problems with a two-environment setup is that it becomes easy to lose track of changes that have been made in the development environment. For example, you may test changes to a provisioning policy without realizing that your tests are being influenced by forgotten changes to some other provisioning policy. After promoting the changes to the only policy that you know about you find that the tests that worked in the development environment fail in the production environment.

This type of problem can be minimized by separating the development and testing environments. This allows the test environment to be maintained in a known state where it contains just the production configuration and the changes that are currently being evaluated for promotion to production. The development environment can be more loosely controlled. It may contain changes that are intended for production, plus other changes that are just experiments that are never intended to be moved out of the development environment.

Large organizations may find that they need multiple development environments. This allows different groups to work independently on new features without affecting each other. With this scenario the different development groups coordinate the use of the test environment. This ensures that the changes being made by the different groups are tested together in the test environment before being promoted to production.

Similarity to the production environment

Ideally, the test environment will be as close to identical to the production environment as is practical. Whether this is really necessary depends on what types of changes you are testing. Testing a change to the escalation period on an approval activity can be done without replicating the one million accounts in the production environment. On the other hand, testing a simultaneous change to the entitlement parameters of five provisioning policies with the same target service should be done with test data that accurately reflects the production environment. Testing such a change in an environment containing five persons and 10 accounts may not identify a problem that will effect 100 of the one million production accounts. Testing in an environment that accurately reflects the production environment can also identify performance problems that would not be seen in a smaller environment.

It can sometimes be difficult to justify creating a test environment with sufficient power and storage to mirror the production environment. Remember that in addition to acting as your test environment, this environment can also provide standby capacity in the event that any production servers fail.

The development environments can contain a much smaller subset of the production configuration. The Identity Manager import/export tools require that objects be imported into an organizational tree that is similar to that from which the objects were exported. For that reason you will want the development environments to at least mirror the portion of the production environment's organizational tree that contains objects such as service, policies, and roles. Portions of the tree that contain only person objects need not be reproduced in the development environments.

So far we discussed only how much of the Identity Manager configuration and data should be reproduced between the production, test, and development environments. You should also consider whether the managed resources themselves should be reproduced in these environments. Will Identity Manager services in the test environment point at your production adapters? Or will you provide testing environments for your managed resources with their own adapters?

Providing a full test environment of a major enterprise managed platform is often difficult, but it does provide the most reliable test results. It is possible to configure your test environment in such a way that its services point to the production adapters, but are unable to send any requests to those adapters (see the section below). This technique is sufficient if the changes that you are testing are not expected to produce any account operations. Even if you are testing changes to a provisioning policy that you expect to create or modify accounts, you will be able to see what operations would have been requested had the test Identity Manager server been able to communicate with the production adapters, but you will not be able to test the effect of those requests. Until the requests are sent to an adapter, you will not know if the managed resource was able to fulfill the request.

Maintaining the environments

The environments cannot be maintained entirely with the import/export tool. This tool cannot export organizational containers, persons, accounts, or service supporting data (groups, for example) in Identity Manager. If you want to copy these objects between environments, you do it manually. If your test environment is mirroring your production environment, then the easiest way to maintain the test environment is by overwriting its directory data with a backup of the production data.

The only data copied between the environments is directory data. There is usually no need to copy any data between the Identity Manager databases or between the installation files. One exception to this might be when you are adding or changing entries in the CustomLabels.properties file. But under no circumstances should you copy all of the installation files between environments. This would result in both environments using the same directory and database servers.

Isolating the environments

There are two ways in which testing activities might have unintended effects on the production environment. There are that steps you can take to prevent this. The least serious of these problems is the sending of e-mail notifications. Users can become confused if your test server is sending notifications of new accounts, pending approvals, and compliance alerts. This can easily be avoided by routing

the test server's notification messages through a mail server that does not forward the messages. The mail server used by an Identity Manager server is defined during the installation process. It can be changed using the runConfig program in the server's bin directory.

A more serious problem is the accidental modification of accounts on the resources managed by the production server. Remember that if the test environment is created by copying the production environment's directory data, then the services in both environments will have the same URLs pointing at the production adapters. If your testing results in account operations, the test server attempts to perform these operations using the production adapters. There are a number of steps that you can take to reduce the risk of this happening:

- ▶ Always change the service's erUrl attributes after copying the production directory data to the test directory.

The update of the service's URLs could be performed easily using IBM Tivoli Directory Integrator. You will want to do this if you have test versions of your managed resources and must change the services to use the test adapters for those resources. But relying on this alone would be risky. There would always be a danger that someone would forget to perform this step before starting testing.

- ▶ Use different certificates in the test and production installations.

This prevents the test Identity Manager server from establishing connections with the production adapters.

- ▶ Isolate the test server behind a firewall that will not pass traffic to the production adapters' ports.

11.4.3 TAA's implementation

TAA has decided to use three environments:

- ▶ Development
- ▶ Test
- ▶ Production

Creating the development and test environments

TAA has had only one Identity Manager installation during the first two phases of the implementation. This has become the production server.

The development and test servers will be installed using different certificates from the production server and using a test mail server. Following the server installations, the test server's directory data will be deleted and replaced by a copy of the production server's data.

The development server starts as a copy of production, but without any persons or accounts. This requires two steps to set up:

1. Copy the organization tree objects from the production directory to the development directory.

This can be done using Directory Integrator or by exporting the objects as Lightweight Directory Access Protocol (LDAP) Data Interchange Format (LDIF) data. The objects that must be copied are the contents of the directory's `ou=orgChart` container and the organization object. The organization object should be copied because any Identity Manager ACIs defined at the organization level are stored as attributes on this object. See *IBM Tivoli Identity Manager Database and Directory Server Schema Reference, Version 5.1, SC27-2413*, for details on the structure of the Identity Manager directory data.

2. Perform an export/import from the production server to the development server using the *export all* option.

The development server will now have the same policies, workflows, and ACIs as the production environment, but will not have any persons or accounts defined.

TAA has decided that it is not feasible for it to provide full-scale test environments for all of its managed platforms. Each platform will have a test environment, but these are not capable of supporting the number of accounts that are maintained in the production environment. This forces TAA to perform some types of testing in its development environment instead of its test environment. Any testing that involves verifying that requests being generated can be successfully processed by the adapters and their managed platforms will be done in the development environment using the test adapters. Any testing that involves verifying that the expected account requests are generated will be done in the test environment where the full set of person and account data is available for testing. The results of this type of testing are checked by looking at the pending requests that are generated by the testing. The requests are never complete since the test server is not able to connect to the production adapters.

TAA's test and promotion procedures

TAA has created an Identity Manager change control board (CCB) that will monitor and control changes being made to the production Identity Manager installation. The CCB is responsible for approving test plans for Identity Manager changes and scheduling the promotion of changes from test to production.

The Identity Manager implementation team works on new features and configurations in the development environment. When they believe that a new feature set is ready for production, they begin the following process:

1. A proposal is given to the CCB describing the features, expected risks, and a test plan. The process continues if the CCB approves the proposal.
2. The test server's directory tree is deleted and then restored from the most recent backup of the production directory server.
3. The objects comprising the new feature set are exported from the development environment and imported into the test environment.
4. Testing is done following the test plan approved by the CCB. The process continues if the testing is successful.
5. The test results are presented to the CCB along with a list of new and changed objects that are promoted to production, and a suggested set of validation tests that should be performed in the production environment after the promotion. The process continues if the CCB approves the test results.
6. An export is done from the test environment using the full export option.

Doing a full export ensures that none of the changes that were just tested will be left out when the changes are promoted to production. It also allows the import function in the production environment to check for conflicts in objects that were not supposed to change.

7. The exported test objects are imported into production.

The import proceeds in two steps. The first step looks for objects being imported that will overwrite existing objects. These objects are compared, and if any differences are found, the user performing the import is asked to specify whether the imported object or the existing object should be preserved.

The administrator doing the import should compare this conflict list with the list of expected changed objects that was submitted to the CCB in step 5. The import should be aborted if any of the detected conflicts are unexpected. This would indicate that objects not covered by the test plan were changed in the test environment or that some objects in the production environment have been modified since the test environment was created.

If the list of conflicts produced by the import agrees with the list of expected changes, then the administrator proceeds with the second phase of the import. This is the phase that creates and modifies objects based on the import data. When this is complete the administrator performs the validation tests that were approved by the CCB.

Archived



Technical implementation: Phase IV

In this chapter we provide information about deploying the role-based access control (RBAC) model. We identify and deploy separation of duty policies and define recertification policies for accounts, roles, and groups.

12.1 Preparing for role-based access control

Implementing role-based access controls and identifying roles that may have conflicts of interest, as candidates for separation of duty (SoD) policies, may be the most time-consuming and complex part of an identity management deployment because it has the potential to affect the entire enterprise, as you will be granting IT system access based on global job roles or functions. See Figure 12-1 for a typical job function to system access mapping.

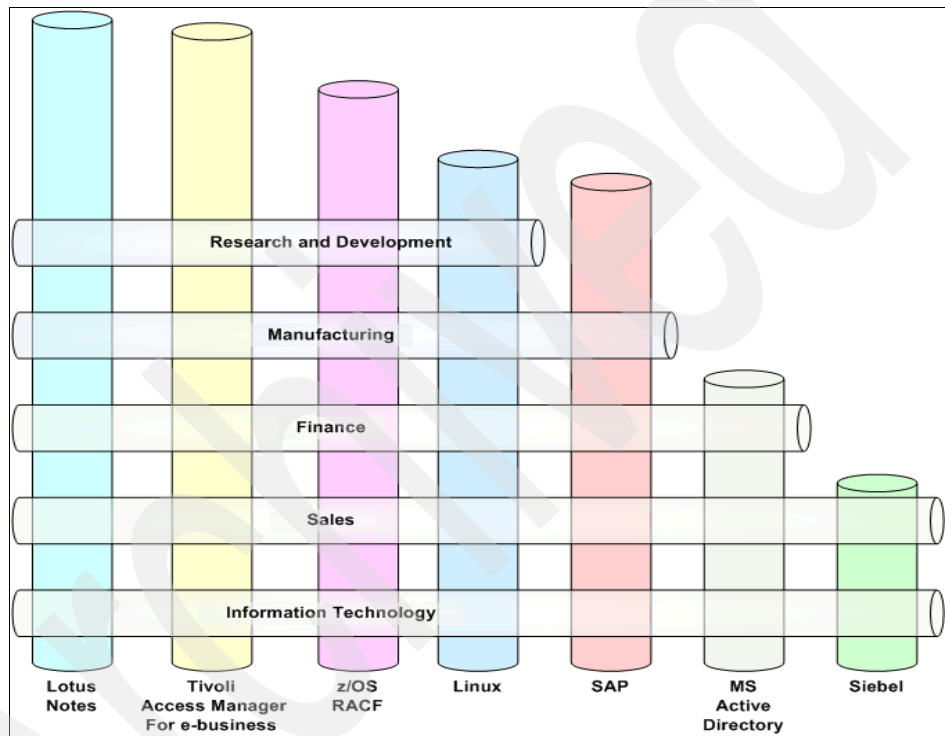


Figure 12-1 Typical job function to system access mapping

It may also yield the largest portion of the enterprise's return on investment (ROI) because the enterprise may use this as an opportunity to:

- ▶ Re-engineer its business processes.
- ▶ Update its security processes.
- ▶ Revise its corporate security policies.
- ▶ Comply with recently enacted legislation.

Unless this is a new business, there will probably be a security infrastructure containing many unordered relationships in place. Most likely you will find the history of using a discretionary access control model (DAC) or a mandatory access control model (MAC), or a combination of both. There may be limited information for determining how user accounts are assigned to groups for establishing access to various managed resources (servers, applications, networks, and so on) based on a person's job function or role.

Figure 12-2 shows a small IT security infrastructure whose contents may not be current. For example, there may be accounts, groups, and managed resources that are no longer used and should be deleted. There may be accounts, groups, and managed resources in need of updating because personnel may have changed jobs. The systems administrators that defined the access controls to these resources may no longer work for the business. More reasons may include the lack of information about the managed resources or people, or complex and outdated group memberships.

Most importantly, there is no obvious information to:

- ▶ Map a person to any account.
- ▶ Map a person's accounts to the managed resources and level of access based on their job function.

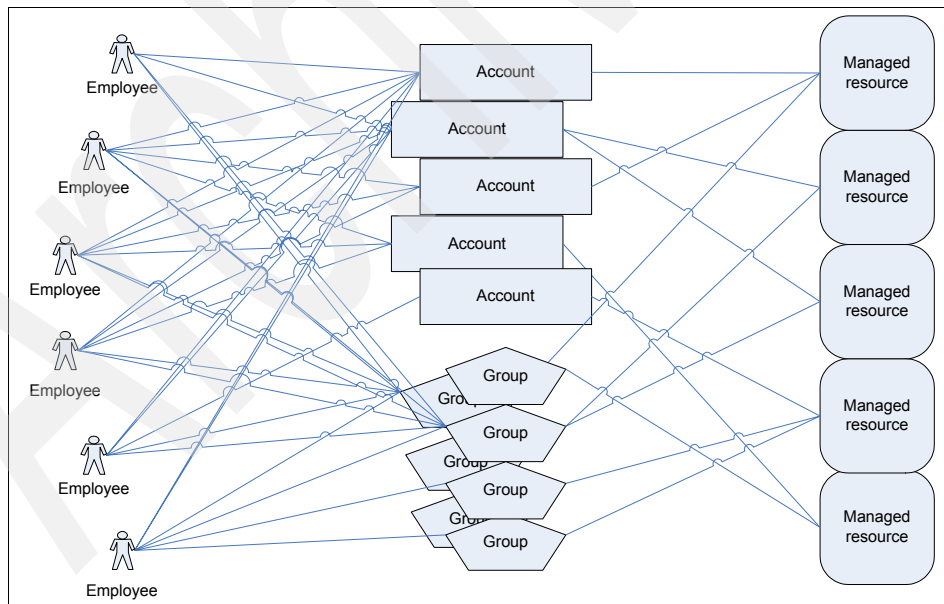


Figure 12-2 Historical security infrastructure

12.1.1 Knowledge gathering approaches

In order to implement role-based access control, you must have a clean underlying security infrastructure. An extensive knowledge-gathering process should be performed by a multiple-discipline team of people to obtain an accurate picture of the information resources that a person must have to perform their job and how they obtain these resources. The team should consist of the:

- ▶ Business owners: Managers from the departments for which roles will be defined plus one or two senior members of their staff
- ▶ Human resources analyst
- ▶ Systems or security administrator for each managed resource
- ▶ Tivoli Identity Manager solution designer
- ▶ Tivoli Identity Manager deployment engineer

This team should be trained to develop the use cases that will become the specifications for defining and testing the RBAC components such as the:

- ▶ Organizational roles
- ▶ Provisioning policies
- ▶ Accesses
- ▶ Service selection policies
- ▶ Workflows and other business processes
- ▶ Separation of duty policies
- ▶ Identifying potential role relationships and role hierarchy structures

Due to the complexity of this task, the team must develop a phased approach to transition the business from its current mode of operation to using an RBAC model without disrupting any business operations. The migration should have little to no impact on the user community.

While gathering data, the team should be looking to identify role relationships that can be used to form hierarchical structures. These hierarchies can be used to define roles at a granular level, while hiding the complexity from higher levels of the configuration. An example is that *business managers* and *executives* can be defined as independent roles, which accurately define their business function. However, these roles can be grouped at a parent level of *manager* when the distinction between business manager and executive is not required. For a more detailed explanation of role relationships and role hierarchy, see 12.1.3, “Defining roles” on page 521.

It is important early on in a deployment to start considering roles that may have conflicts of interest and require *separation of duty* (SoD) policies to be defined. Only static organizational roles can be managed by separation of duty policies, so any potential SoD candidate roles must be static roles.

The team should leverage the change management and configuration management disciplines to avoid wasting time and resources. For example, if there is an ongoing server consolidation project, the team should focus its efforts on the consolidated or target server versus the source servers that will be eliminated.

There are two frequently used knowledge-gathering approaches:

- ▶ The top-down approach
- ▶ The bottom-up approach

A description of each approach, its advantages, and its disadvantages is shown in Table 12-1.

Table 12-1 RBAC knowledge gathering approaches

Approach	Advantages	Disadvantages
Top-down: Add new roles into an existing security infrastructure.	<ul style="list-style-type: none"> ▶ Speed ▶ Acceptable for small-scale and non-complex environments 	<ul style="list-style-type: none"> ▶ The discovery process may be incomplete because you may not have an up-to-date inventory or a thorough understanding of all servers and applications. ▶ Risk of disrupting.
Bottom-up: Restructure the security infrastructure and build new roles on it.	<ul style="list-style-type: none"> ▶ Building on your current base while removing outdated information ▶ Best for addressing critical and complex needs 	<ul style="list-style-type: none"> ▶ Lack of speed. ▶ Extensive analysis is required. ▶ Risk of disrupting business operations.

Each approach has blindspots, and to overcome them we suggest using a third approach that we are calling the hybrid approach. It combines both the top-down and the bottom-up approaches in order to validate the results of each. It should also answer the following questions:

- ▶ What are the characteristics of the business?
 - Is there a high employee turnover ratio?
 - Do employees change jobs frequently?
 - What is the ratio of accounts per person?
 - What is the ratio of manager to non-manager?

- ▶ What is the role assignment criteria?
 - Job code
 - Location
 - Manual assignment or self-request
 - Other
- ▶ Where does the data exist for assigning persons to roles?

For example, a job code may be stored in a directory or file that is accessible to Tivoli Identity Manager. However, a team assignment may not be stored.
- ▶ What are the characteristics of the operational Tivoli Identity Manager instance?
 - What is the structure of the organization tree?
 - What is the largest number of users for a given organization unit?
- ▶ Does the role need to be considered for inclusion in an SoD policy?
- ▶ What kind of role should be used, static or dynamic?
- ▶ Could the role benefit from being defined as part of a role hierarchy?
- ▶ What are the performance ramifications if a dynamic role is selected?
 - How often will a user's attributes be updated?
 - How often will an associated provisioning policy be updated?
 - How many persons are assigned to each role?
 - May updates be scheduled during off-peak hours, nights, or weekends?
- ▶ Who creates the role?
- ▶ Who maintains the role?
- ▶ Which accesses should be made available to the user?
- ▶ Who owns the access?
- ▶ Which business processes should be applied to the user's accounts and accesses?

From these interviews, the knowledge-gathering team may prepare use cases. As they analyze these use cases and drill down into the underlying details by answering the who, what, when, where, why, how, and how many questions, patterns should emerge for identifying:

- ▶ Requirements for data access
- ▶ Requirements for entitlements to managed resources
- ▶ Requirements for business processes (such as approval and recertification) for access data and managed resources

During this process, the team is working to reduce the number of Tivoli Identity Manager configuration items to be implemented. First and foremost are organizational roles, which may be shared across different departments or different job codes. When appropriate, some of the organization roles can be structured into role hierarchies. This can be used to maintain distinct roles while hiding complexity from higher level configuration items, such as provisioning policies. The team will be working to find the lowest common denominator. This process may also apply to other configuration items such as ACIs and Tivoli Identity Manager groups. In the database world, this is called a normalization process. For Tivoli Identity Manager this is a necessity in order to simplify Tivoli Identity Manager administration and improve Tivoli Identity Manager performance.

Archived

Figure 12-3 shows a desired outcome.

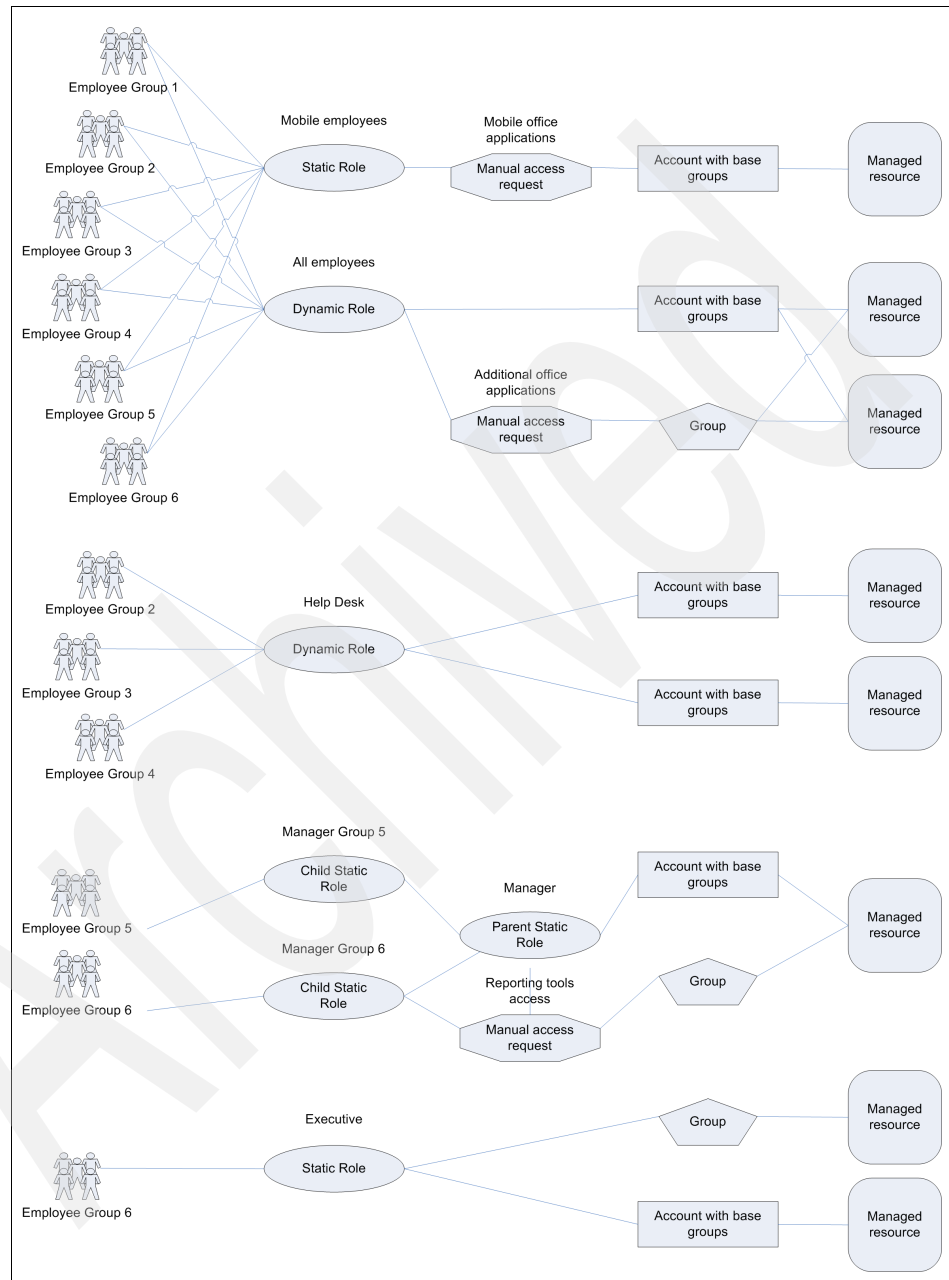


Figure 12-3 Sample role mapping

An ideal role-based access control configuration results in account management being automated based on a person's enterprise characteristics. When automation should not occur, either because account entitlements are request based or should be manual by policy (for example, with an approval process attached to them), a request-based system should be implemented, in line with role-based access control configuration.

In this example, the *all employees* role entitles everyone in the company access to multiple resources (Tivoli Identity Manager managed services) that are needed for all jobs such as e-mail, Web portal access, and self-service applications for benefits and activities. A manual access request is available for users who require an additional set of applications or privileges. Accesses mapped to managed resource groups can be subjected to an approval and recertification processes to ensure that only employees with a valid business reason are granted these accesses.

The help desk role is more narrowly defined and it grants only help desk employees access to an additional subset of resources (Tivoli Identity Manager managed services) that they require to perform their work.

The managers role is also more narrowly defined by two child roles; however, the complexity is hidden by the parent role definition. It entitles the management employees access to an additional subset of resources (Tivoli Identity Manager managed services), as well as an optional access to a managed resource group granting them an additional set of accesses.

The last role is the most restrictive and it entitles an employee at the executive level access to an additional set of resources (Tivoli Identity Manager managed service). This sensitive role can be subjected to an approval when assigned to individuals.

12.1.2 Setting expectations

The most successful Tivoli Identity Manager deployment teams use a multiple-phase approach. As previously mentioned, they form a multiple-discipline team of business, IT, and security experts. Ensure they receive Tivoli Identity Manager training, and build the Tivoli Identity Manager systems instances.

This team selects a small business unit for a phase 0 or pilot phase. The business unit selected is based on the number of employees, their needs for IT resources, and the least amount of impact to the business.

During the pilot phase, the team learns to prepare use cases and perform the data mapping exercises that are required to create the Tivoli Identity Manager objects for implementing RBAC, role hierarchies, and, where appropriate, SoD policies. There are lessons learned, but the important aspect is that they are developing a repeatable methodology that solves their specific problems. This methodology, along with the tools and intellectual capital, is refined in later phases as more business units are added.

An Tivoli Identity Manager deployment scales two ways:

- ▶ Vertically by adding more people
- ▶ Horizontally by adding more services

A phased approach is essential because the load on the Tivoli Identity Manager servers increases as the workload increases. The team must monitor the servers and the network to ensure that the workloads match the planning assumptions for the hardware and the network. The team should also schedule quarterly review meetings to assess the need to install corrective or preventive service (for example, fix packs) to any of the components forming the Tivoli Identity Manager software stack.

The team should also be developing and testing the procedures for maintaining steady state operations, such as:

- ▶ Operator procedures
- ▶ Monitoring procedures
- ▶ Data backup, recovery, and restore procedures
- ▶ Server backup, recovery, and restore procedures
- ▶ Network backup, recovery, and restore procedures
- ▶ Disaster planning

These activities represent a significant front-end investment that will yield significant returns in the later phases (measured in cycle time reductions), provided that the multiple-discipline team is kept intact and the team members possess the required skills.

The key to success is obtaining executive-level project sponsors who view the security of IT resources as mission critical. They must have the patience and funds to see the project through completion. A phased approach with an incremental funding model is a proven and successful technique. The funding for the next project phase is not released until several milestones are reached to indicate that the current project phase is successful.

The multiple-phase deployment model can drive incremental resource additions for hardware, software, network capacity, head count, and so on. Managers can leverage this model to develop training plans for their staff. In cases where

automation reduces a person's workload, the manager can prepare new assignments for that person.

The project sponsors should prepare monthly executive briefings to report progress. There should also be frequent briefings to the leaders of the business units affected by the Tivoli Identity Manager deployment. Some companies developed an internal Web site to keep their business community informed of their successes and to deliver Tivoli Identity Manager training materials.

12.1.3 Defining roles

Before you define any roles, an understanding of role relationships and hierarchical role structures will help you in designing effective roles. The first principle to be aware of is *role relationships*. Within role relationships, privileges are passed from parent to child, as shown in Figure 12-4.

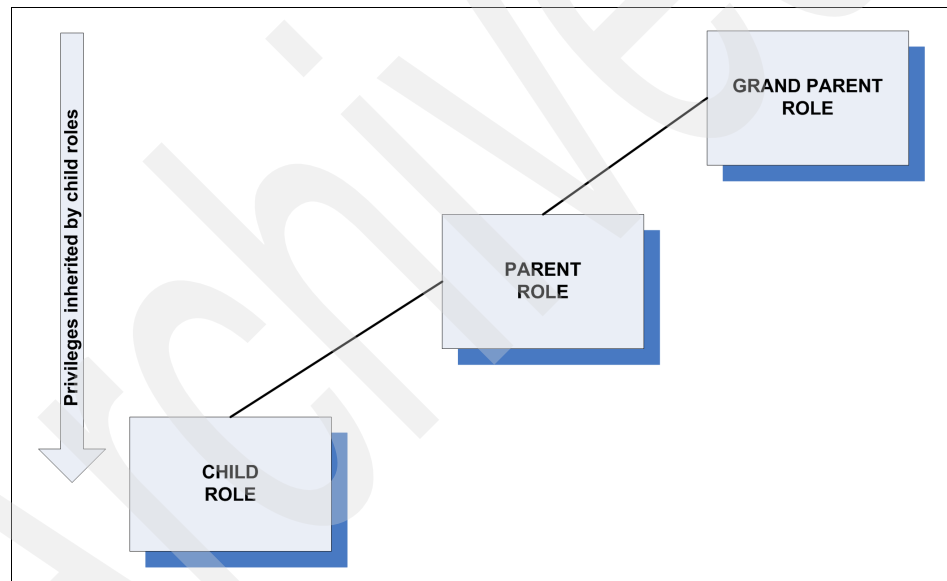


Figure 12-4 Role relationships

The next principle is to know when to build *role hierarchies*. Hierarchical role structures are effective when:

- ▶ Granularity of the roles should be maintained, but can be grouped together to hide complexity from higher levels of the configuration.
- ▶ Management of role membership will be delegated to different administrators, but you wish to maintain one higher level definition.

Role hierarchy structures may lead to more child roles being defined, which will add to the administrative impact of maintaining the roles. So before creating hierarchical role structures, consider who will manage the roles and whether the granularity is needed.

The last principle to consider is that roles forming part of hierarchical role structures must be *static organizational roles* only. Dynamic roles are not supported as part of hierarchical role structures.

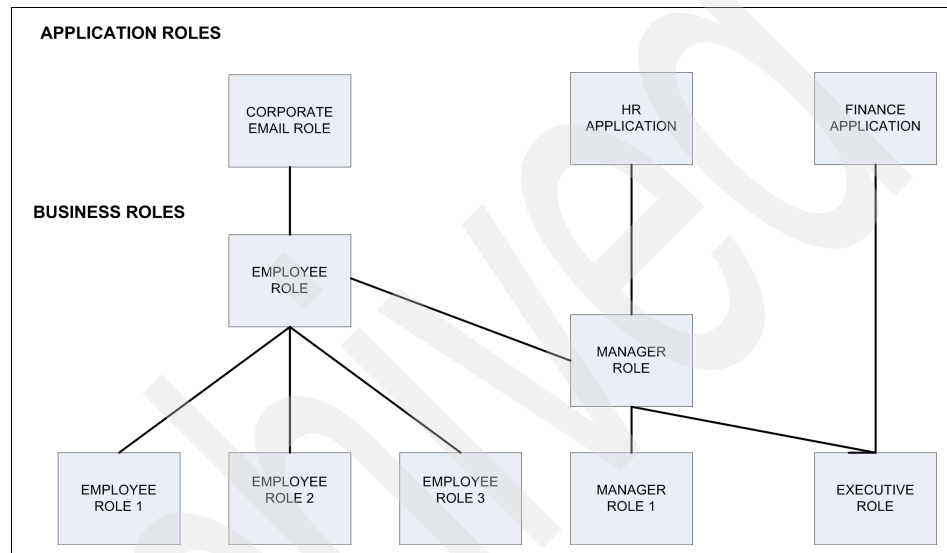


Figure 12-5 Role hierarchy structure

Figure 12-5 shows a simple but representative role hierarchy. Note the following advantages of these structures:

- ▶ Everyone has access to the corporate e-mail role by way of the role relationships.
- ▶ A manager inherits from the employee role of which it is a child. This gives managers access to everything to which employees have access.
- ▶ Executives inherit from the manager role of which it is a child. This gives executives access to everything to which a manager has access, plus access to everything the employee role provides.
- ▶ Only the executive role has access to the finance application role.

Figure 12-5 also introduces *role classification*. When defining roles, you can define a role classification. The default role classifications are *business role* and *application role*. Additional role classifications may be added if required by using Tivoli Identity Manager customization.

In order to manage role-based access control using Tivoli Identity Manager, it is crucial to have a good understanding how role-based access control affects account management on the target system. Role definitions are also a prerequisite before separation of duty (SoD) policies can be defined.

The first piece of information required is which identities should have accounts on target systems. Roles grant permissions for Tivoli Identity Manager identities to have accounts on target systems. Conversely, an account will be non-compliant if its owner does not have any role allowing him to have an account on the corresponding target system.

This information should then be refined to define which identity subset should have access to specific account entitlements and groups. Since Tivoli Identity Manager permissions are cumulative, this can be done in two ways:

- ▶ The first way is to separate those identity subsets into completely separate sets, based on their different sets of permissions.
- ▶ The second way is to define a global set of permissions that apply to all users, and define below cumulative sets of permissions specific for each sub-groups of users, and so on. This second way is generally preferred, as it reduces the number of roles to be managed and reduces duplication of functionality. Since large enterprises may have a very large number of different organizational roles, ease of implementation and flexibility is strongly encouraged.

Once information has been collected about distinct groups of users, this information should be implemented in Tivoli Identity Manager as organizational roles and, where appropriate, defined in hierarchical structures. Organizational roles are organizational identity information snippets that are associated with individual identity records. Multiple roles can be associated with individual identities, roles can have multiple members, and static organizational roles can have parent child relationships.

An additional consideration when implementing roles is how they will be assigned to users. Tivoli identity Manager organizational roles can be assigned to individual identities in one of three ways:

- ▶ Through manual assignment by an administrator or external process, such as an identity feed. The role is then referred to as *static*.
- ▶ Through a manual access request by the user himself or another person with the correct permissions. The *static* role is then mapped to a role access.
- ▶ Automatically whenever the identity information satisfies the membership criteria. Conversely, the role membership is removed when the identity information no longer satisfies the membership criteria after an update. The role is then referred to as *dynamic*.

Note that dynamic roles can also be mapped to accesses. However, these accesses can only be viewed, not requested, owing to the nature of the role.

Figure 12-6 illustrates the various organizational role membership assignments.

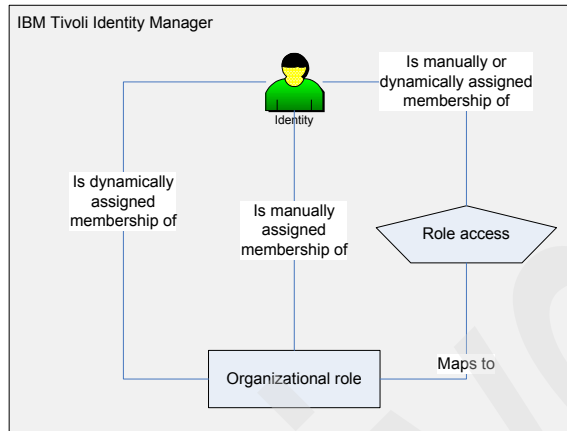


Figure 12-6 Tivoli Identity Manager role membership mapping and assignment

The users-privileges information, or organizational roles-entitlements information as it will be implemented in Tivoli Identity Manager, is important, as it will be used by Tivoli Identity Manager to determine whether user accounts are compliant or non-compliant, and in the latter case which corrective action should be taken. This is referred to as role-based access control. The information collected should therefore be as accurate as possible.

Account entitlement information is then associated with organizational roles using provisioning policies. Provisioning policies enforce which set of account entitlements a set of roles can have. More information about provisioning policies can be found in 12.1.4, “Defining provisioning policies” on page 528. Figure 12-7 on page 525 illustrates how identities, roles, provisioning policies, and target system accounts relate to each other.

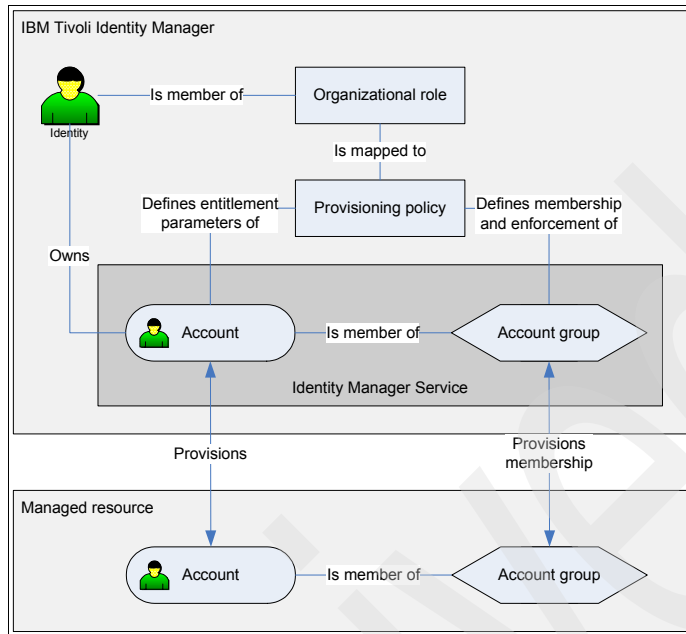


Figure 12-7 Tivoli Identity Manager and role-based access control

Figure 12-7 shows how Tivoli Identity Manager uses role-based access control to manage target system accounts. Several important aspects are shown in Figure 12-7:

- ▶ Tivoli Identity Manager services are a local representation of a target system's, or managed resource.
- ▶ Reconciliation collects information about which accounts and related information, such as group memberships, exist on the target system. It then stores it into the Tivoli Identity Manager system under the managed resource's corresponding service.
- ▶ Identities in Tivoli Identity Manager are mapped to organizational roles, which are themselves mapped to provisioning policies. Remember that static organizational roles can potentially include child roles.
- ▶ Provisioning policy entitlements determine what the account should look like, including group membership. The information is then provisioned to the managed resource.

By analyzing Figure 12-7 on page 525, it is easy to see that by using Tivoli Identity Manager the mapping of identities to entitlements follows the path shown in Figure 12-8. Though not explicitly shown, the organizational role in Figure 12-8 could be dynamic or static, and in the case of a static role, it could form part of a role hierarchy.

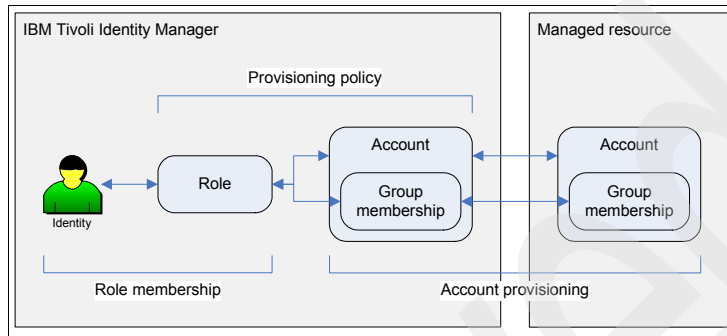


Figure 12-8 Identity to entitlement mapping using Tivoli Identity Manager

Role ownership and approval

When defining roles, it is possible to define role approval in Tivoli Identity Manager and to define who owns a role. Identity Manager allows role owners to be defined as a person or another role. Allowing roles to own other roles is useful when role membership approvals need to allow multiple approvers. This shares the responsibility of approval.

Tivoli Identity Manager allows an owner to be defined for the organizational role. Access control items can be set to enable a role owner to have specific permissions to manage the roles that they own.

Figure 12-9 shows how a person operation workflow can be customized to perform organizational role approval, based on role ownership. A number of workflow extensions have been defined to enable role approvals.

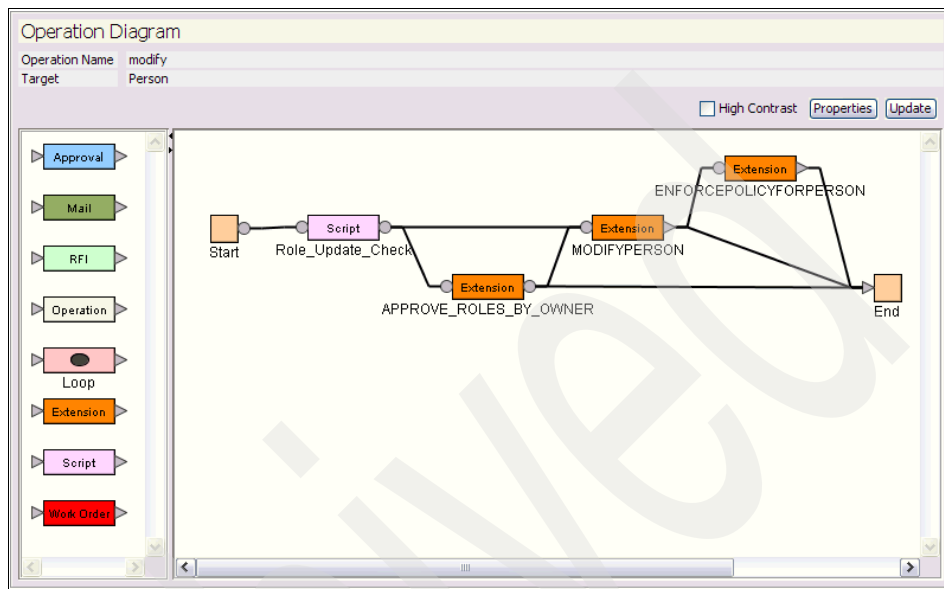


Figure 12-9 Person operation workflow customized for role approval

Role classification

Tivoli Identity Manager allows for the classification of roles. The default values are:

- ▶ Business role
Business roles encompass the kind of job that a person does.
- ▶ Application role
Application roles encompass the kind of access that the person requires.

Figure 12-5 on page 522 shows an example of business and application roles. Additional role classifications may be defined by updating the Tivoli Identity Manager data file `enRole.properties`, as shown in Example 12-1. This example defines an additional classification of *control*. You will need to also define the label `role.classification.control` in the relevant *CustomLabels.properties* file for your supported languages.

Example 12-1 enRole.properties file, role classifications

```
enrole.role.classifications=role.classification.none \  
                             role.classification.application \  
                             role.classification.business \  
                             role.classification.control
```

12.1.4 Defining provisioning policies

Large enterprises may have a large number of organizational roles and a large number of target systems, each with a large number of different accounts with varying combinations of permissions and group memberships. Mapping these using provisioning policies may therefore require careful planning in order to reduce complexity and the impact of organizational changes onto ongoing configuration.

There are several things to keep in mind when designing and configuring provisioning policies:

- ▶ A single person may have multiple roles, and also possibly be a member of one or more role hierarchy structures, which can themselves be mapped to multiple provisioning policies. These, in turn, may contain entitlements for multiple services, which can overlap from one policy to another.
- ▶ Provisioning policies are assigned priorities so that in case of entitlement conflicts when multiple policies are applied to a single individual account, one provisioning policy's entitlements are enforced rather than another. This affects entitlement parameter attributes for which a *policy join behavior* has been set to *priority*, referring to the priority of the corresponding provisioning policy. Other policy join behavior options include *append*, *and*, and *highest*. Policy join behavior for account attributes can be set in the Tivoli Identity Manager administrative console's system configuration task panel.
- ▶ A workflow can be configured for each provisioning policy entitlement. These workflows are executed prior to the creation of the account corresponding to the provisioning policy's entitlement. They are normally used to define account approvals of requests for information. If the account already exists when the policy is applied (such as when the account is modified or the owner's person record is updated), the workflow is not executed.

When multiple provisioning policies all have entitlements for the same service, and they have different workflows, the workflow belonging to the policy with the highest priority is executed (unless the account already exists, in which case no workflow will be executed).

- Provisioning policy entitlement parameters define how entitlement attribute values should be populated. They can be defined using constant values, JavaScript that can access information from the account's owner, regular expression, or a *null* value. Attributes can be either single valued or multi-valued. Figure 12-10 shows sample provisioning policy entitlement parameters.

The screenshot shows the Tivoli Identity Manager web interface. The breadcrumb navigation is 'Manage Policies > Manage Provisioning Policies > Entitlement Parameter'. Below the breadcrumb, there is a text instruction: 'Select one or more provisioning parameters that you want to change and click Change, or select Create to view a list of attributes from which you can select to add a new attribute. To remove an attribute, select the attribute, and then click Delete.' There are three buttons: 'Create', 'Change', and 'Delete'. Below this is a table with the following data:

Select	Name	Template value	Enforcement...	Value Type
<input type="checkbox"/>	Distinguished Name	return "cn="+subject.getProperty("cn")[0]+"*,ou=People,o=TAA,c=us";	Default	JavaScript
<input type="checkbox"/>	Description	Account created by Tivoli Identity Manager	Default	Constant Value
<input type="checkbox"/>	Last name	return subject.getProperty("sn")[0];	Default	JavaScript
<input type="checkbox"/>	Full name	return subject.getProperty("cn")[0];	Default	JavaScript
<input type="checkbox"/>	Single Signon Capability	true	Default	Constant Value
<input type="checkbox"/>	Group Membership	SecurityGroup	Mandatory	Constant Value

At the bottom of the table, it says 'Page 1 of 1', 'Total: 6', 'Displayed: 6', and 'Selected: 0'. There are 'Continue' and 'Cancel' buttons at the bottom of the page.

Figure 12-10 Provisioning policy entitlements

- Enforcement types can be applied to entitlement parameters, which affects how they can be accessed by users and how they accumulate across multiple provisioning policies. These include:
 - Default enforcement type (available for both single-valued and multiple-valued entitlement parameter attributes): The configured values will be assigned to the corresponding attribute if no other values are present. Any other values are valid.
 - Mandatory enforcement type (available for both single-valued and multiple-valued entitlement parameter attributes): The configured values are required to be assigned to the corresponding attribute. If an entitlement parameter has only *mandatory* and *optional* enforcement values, no value other than these is allowed.

- Optional enforcement type (available only for multiple-valued entitlement parameter attributes): The configured values are allowed for the corresponding attribute.
- Excluded enforcement type (available only for multiple-valued entitlement parameter attributes): The configured values are not allowed for the corresponding attribute.

Figure 12-10 on page 529 includes sample enforcement types for entitlements parameters.

Mapping roles, provisioning policies, and entitlements

A large enterprise may have a large number of target systems and business roles. They may, in turn, require a large number of Tivoli Identity Manager organizational roles, services, and provisioning policies to be set up to manage these. There are various ways to configure Tivoli Identity Manager to set up roles and provisioning policies. Because of this, it is important to keep the complexity of Tivoli Identity Manager's design and configuration to a minimum.

Figure 12-11 shows a sample set of identities, either individuals or groups of individuals, mapped to the account and groups that they are entitled to have based on their job roles. Each has different accounts that span multiple target systems.

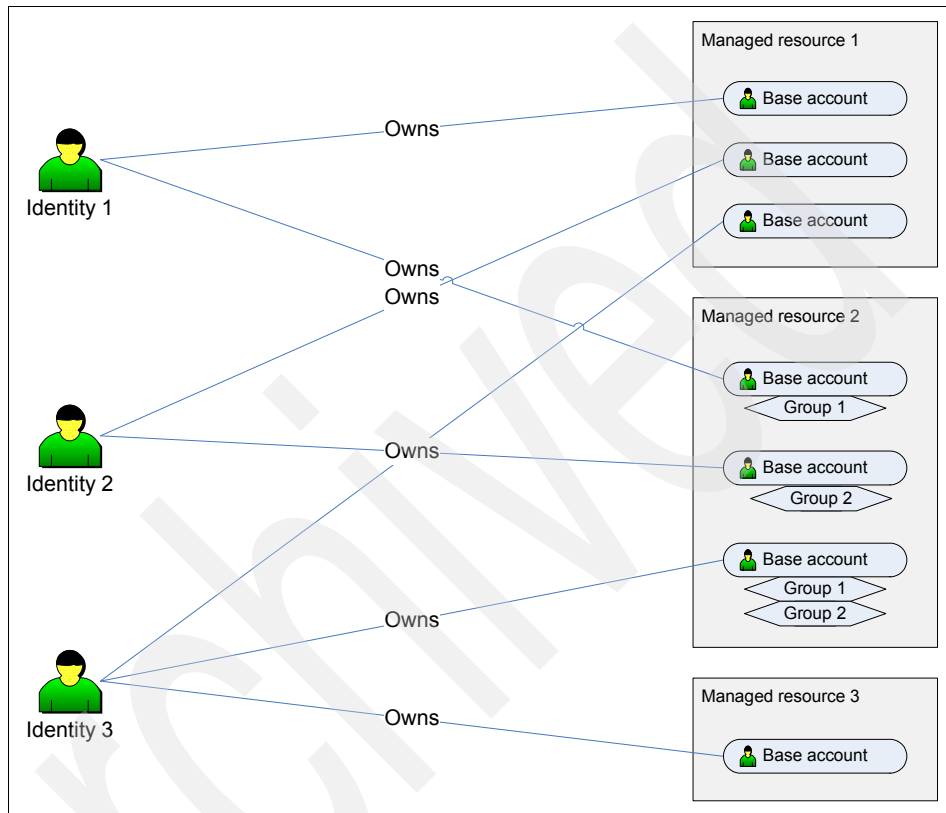


Figure 12-11 Sample identity account mapping

One approach that is frequently considered to map roles, provisioning policies, and entitlement is *one-to-one* mapping based on account entitlements. This approach directly maps provisioning policies and roles to account entitlements. Figure 12-12 shows a sample one-to-one Tivoli Identity Manager role, provisioning policy, and entitlement configuration applied to the sample identity account mapping shown in Figure 12-11 on page 531.

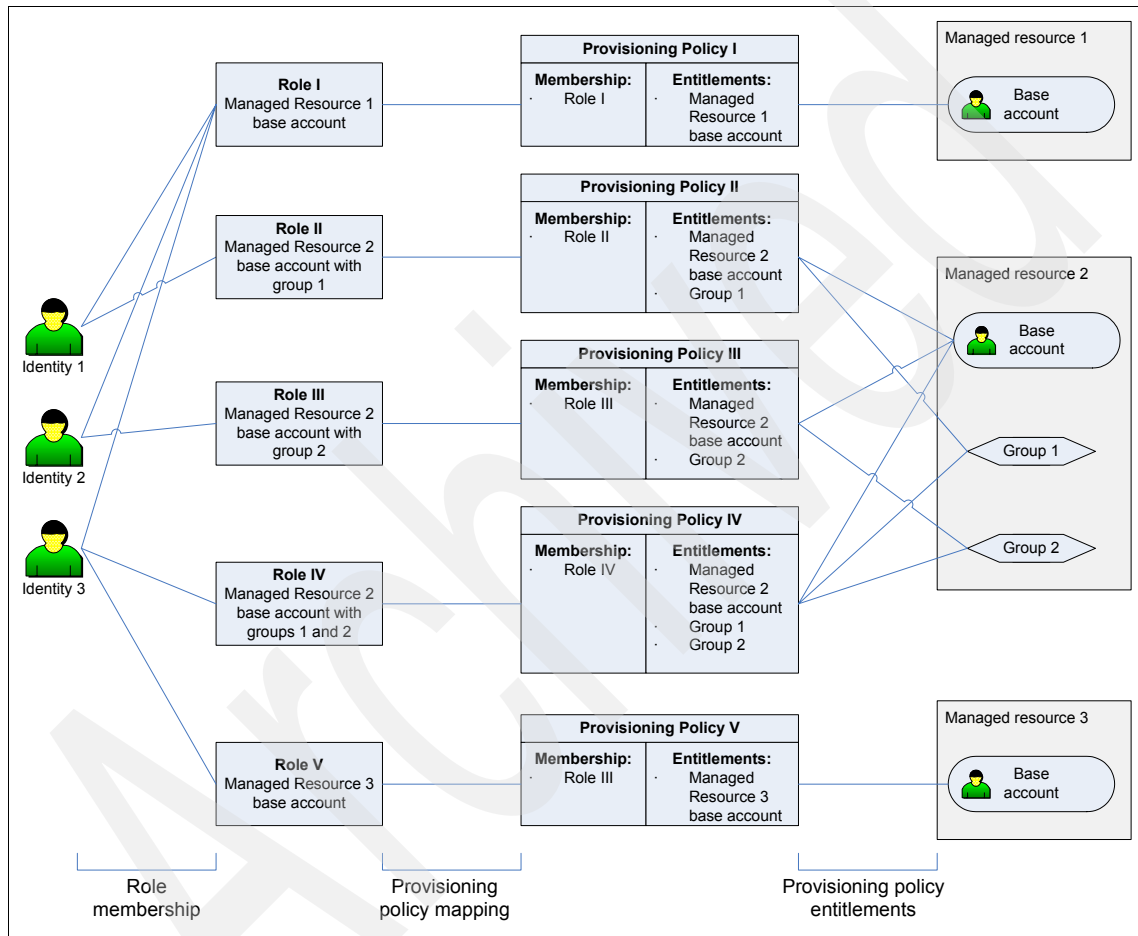


Figure 12-12 Sample one-to-one role, provisioning policy, and entitlement mapping

While this approach may appear simple to manage, in practice it creates a model that is not driven by identities and may result in a structure with complex roles and policies and have overlapping information.

Another approach is to design the role, provisioning policy, and entitlement mapping using the principle of the *lowest common denominator*. You also have the option to use role hierarchy and role relationships to define the common lowest denominator while still maintaining distinct role definitions. Refer back to Figure 12-5 on page 522. This is a good example of the *corporate e-mail role* being defined as the lowest common denominator while still maintaining a meaningful role structure. Remember, the principle of lowest common denominator can be applied to a single role or a role hierarchy structure. This approach can be implemented with the following steps:

1. Determine the largest number of common configuration elements found for a set of identities or entitlements. In the case of provisioning policy entitlements, this may span multiple Tivoli Identity Manager services and may include common attribute value mapping and groups. In the case of organizational roles, it should include identities with a similar set of responsibilities, accesses, or geographical locations. Define your first role, role hierarchy or policy with this set of configuration elements.
2. Focusing on the remaining configuration elements, find again the largest number of common identities or entitlements configuration elements. Define a new role, role hierarchy or policy with this set of configuration elements.
3. Repeat steps 1 and 2 until all configuration elements have been entered into a role or policies.

Figure 12-13 shows a sample Tivoli Identity Manager role, provisioning policy, and entitlement configuration that follows the principle of the lowest common denominator.

Note: Any of the roles shown in Figure 12-13 could be replaced by a role hierarchy structure if there is a business need to maintain distinct roles.

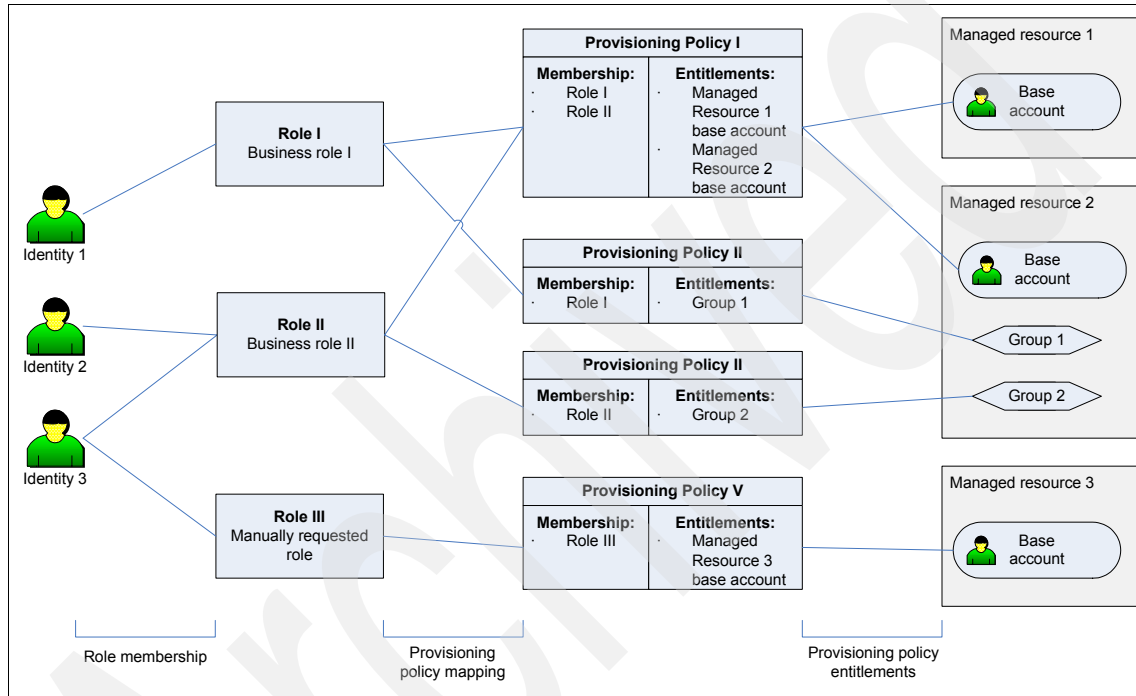


Figure 12-13 Sample lowest common denominator role, provisioning policy, and entitlement mapping

We recommend this approach as it encourages simpler, more flexible Tivoli Identity Manager configuration with minimal overlap. Also, the organizational role design follows the role-based access control model, which enables identities to manage account entitlements. A downside, however, is that this approach may result in roles and provisioning policy entitlement not fully mapping to one another. These may consequently need some adjustments to align them.

In the real world, single approaches can rarely be applied to enterprise environments. Due to their variety, size, and complexity, it is highly unlikely that a single method will be able to efficiently account for all identity and account management needs. Nevertheless, we recommend the lowest common denominator approach to create the bulk of role, provisioning policy, and entitlement mapping in an enterprise.

12.1.5 Defining accesses

In cases where identity information is not enough to determine whether individual entitlements should be granted to a user, either because the process requires being assigned manually or having a business process, such as an approval, attached to it, accesses can be implemented to simplify request-based roles and entitlement management.

Accesses can also be mapped to dynamically assigned roles and entitlements. However, these depend on logic rather than manual assignment.

The list of accesses available to a user depends on his permissions. For example, if a person is not entitled to have a role mapped as an access by a provisioning policy, this access will not be available to him.

Role accesses

Tivoli Identity Manager organizational roles can be exposed as is using an access definition. Both static and dynamic roles can be exposed as accesses. However, only static roles mapped as accesses can be added or removed manually.

Account group accesses

Account groups can also be exposed as accesses. In contrast to role accesses, these can have an alternative display name and description to make them more meaningful to users, and may have approvals attached directly to them. Role accesses are mapped and assigned as shown in Figure 12-14.

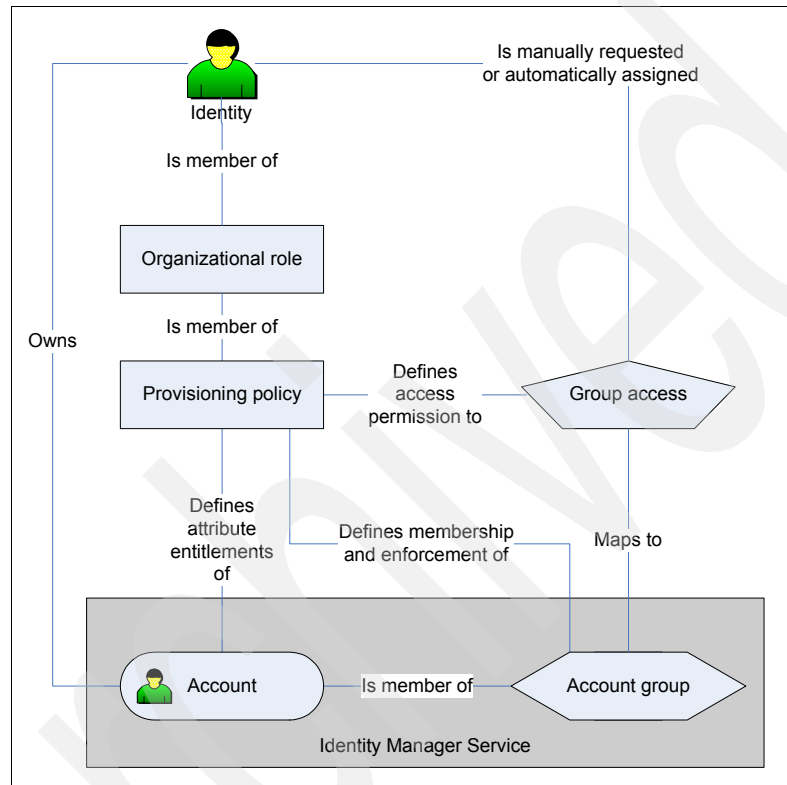


Figure 12-14 Tivoli Identity Manager group access mapping and assignment

12.1.6 Defining separation of duty policies

A separation of duty (SoD) policy is a policy that manages conflicts of interest between mutually exclusive roles. Within these policies, you define two or more roles and the number of the roles one person may be a member of at any one time. If the rule is broken, this would be a conflict of interest and trigger a policy violation.

Figure 12-15 on page 537 shows a simple separation of duty policy example where the separation between *flight operations role* and *aircraft maintenance role* is defined. This enforces a policy where you may only have update access

rights on flight operations or aircraft maintenance but not both. This stops any conflicts of interests, for example, flight operations will not be permitted to remove an aircraft maintenance check. Likewise, aircraft maintenance will not be able to change flight operations if there is a maintenance overrun. Both parties may read each others data, but not change it directly themselves.

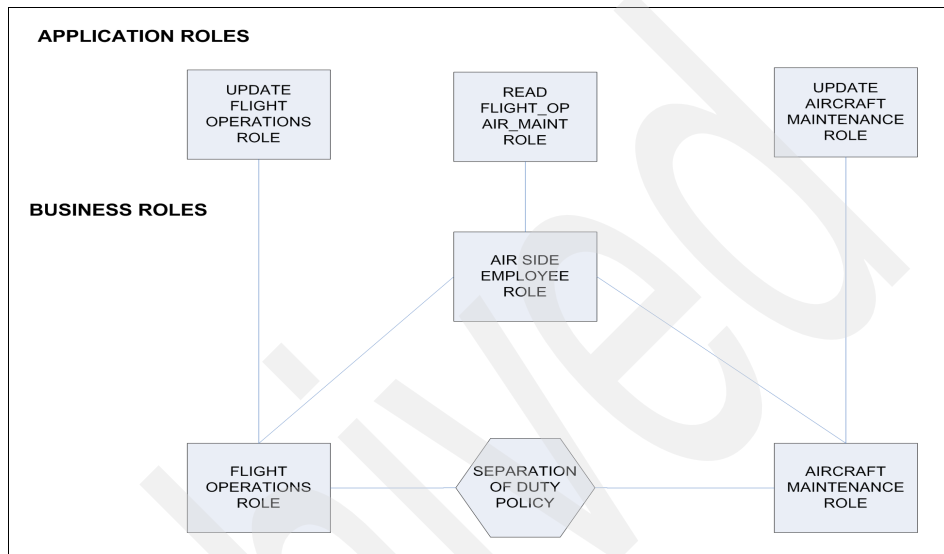


Figure 12-15 Simple separation of duty example

During your data gathering and planning, you should have identified candidate roles that may require separation of duty policies. One approach to identifying roles that may have conflicts of interest is to create a role matrix and define the roles on both the x and y axis. The following steps can be used to build a candidate list of roles for inclusion in separation of duty policies:

1. Build your role matrix.
2. Mark roles that have potential conflicts of interest.
3. Carry out a risk assessment:
 - What is the threat and potential impact?
 - Is the risk already being managed elsewhere in the business?
 - Can the risk be better managed elsewhere in the business?
 - Can the risk be accepted (if accepted, document your reasons)?
4. Clearly identify the roles that are in conflict.
5. Document the conflict for use in the SoD policy description.
6. Identify who should own the policy and approve any exemptions.
7. Add the role to the list of roles to be defined within SoD policies.

Figure 12-16 shows an expanded example where role hierarchy, organizational roles, provisioning policies, and managed systems are shown in context to each other.

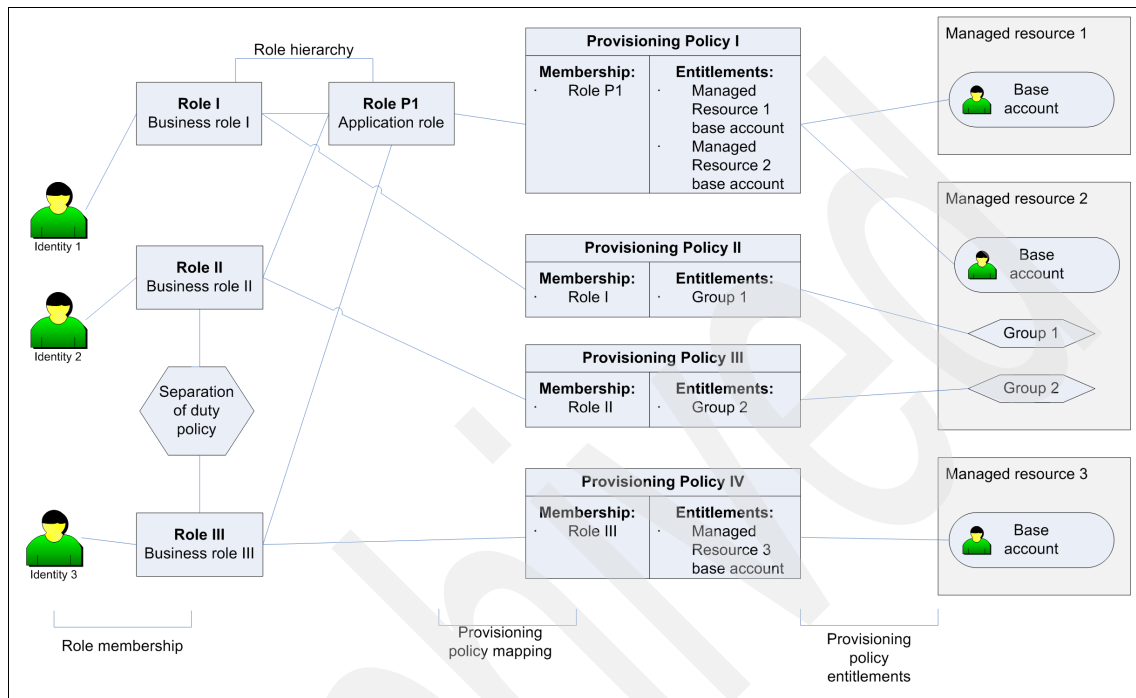


Figure 12-16 Separation of duty policy in context with provisioning policies

12.2 Requirements

This section provides information about how the functional requirements identified in 8.2, “Functional requirements” on page 325, and mapped to this phase are satisfied.

Table 12-2 on page 539 shows the functional requirements versus implementation steps in this phase.

Table 12-2 Functional requirements mapping

Functional requirements	TAA's deployment requirements
C. Common values are entered automatically.	Roles and provisioning policies
F. Allow delegation of approval responsibilities.	Roles and provisioning policies
G. Support collaboration by multiple approvers.	Roles and provisioning policies
H. Remind approvers of waiting requests.	Roles and provisioning policies
I. Escalate ignored requests.	Roles and provisioning policies
K. Automatically add and remove accounts and access rights when a user changes job role.	Roles and provisioning policies
R. Account owners or their managers will be periodically asked to certify their continuing need for their accounts and access rights.	Certification process
S. Accounts and access rights that are not certified are disabled or removed.	Certification process
CC. Based on a defined policy, manage roles that have conflicting interests.	Separation of duty policies

12.3 Design considerations

There are several design considerations for us to observe:

- ▶ Set realistic expectations about migrating to a role-based access control model. It is a complex and time-consuming undertaking because you are not only solving an IT security problem, but you may also be re-engineering business and security processes.
- ▶ Deploy in multiple phases.
- ▶ Start with a small scope pilot organization and build expertise. Keep it simple.
- ▶ Share information. Broadcast successes in terms of impact to the business, such as cost savings or cycle time reductions.
- ▶ Be sensitive to multiple agendas. Not all business units or systems administrators will be eager to participate.

- ▶ Constantly look for common elements in each role and combine them where possible. Contain role propagation.
 - ▶ Where roles cannot be consolidated to the lowest common denominator, look to define role hierarchical structures.
 - ▶ Consider whether you require separation of duty policies to be defined during data gathering and role discovery.
 - ▶ Monitor Tivoli Identity Manager performance as each new role is activated.
- The use of dynamic roles increases Tivoli Identity Manager overhead. For example, when the policy is updated, Tivoli Identity Manager verifies all accounts affected by the policy.

12.4 TAA implementation

The first step of the implementation is the mapping exercise to match a person to their work location and job code. These three attributes represent the RBAC foundation for TAA.

12.4.1 Location codes

Tivoli Austin Airlines locations are listed in Table 12-3. The default locale setting is English unless it is overridden by an entry in the table.

Table 12-3 TAA locations

Location	Description
RW	Regional center west (San Francisco)
RA	Regional center austin including the corporate headquarters and the central IT data center
RE	Regional center east (New York)
SEA	Customer service center (Seattle)
SFO	Customer service center (San Francisco)
LAX	Customer service center (Los Angeles)
AUS	Customer service center (Austin)
DEN	Customer service center (Denver)
STL	Customer service center (Saint Louis)
DET	Customer service center (Detroit)

Location	Description
JFK	Customer service center (New York)
RDU	Customer service center (Raleigh)
MEX	Customer service center (Mexico City)

12.4.2 Employee codes

Tivoli Austin Airlines employee codes are listed in Table 12-4.

Table 12-4 TAA employee codes

Employee code	Description	Requirements
1000	Corporate executive	<ul style="list-style-type: none"> ▶ Approve regional executives' requests. ▶ Read access to all corporate, financial, and operational data.
2000	Regional executive	<ul style="list-style-type: none"> ▶ Approve regional managers' requests. ▶ Read access to all financial and operational data for their specific region.
3000	Regional ground services manager	Approve requests to regional ground services applications for their specific region for personnel in their region.
3100	Regional ground services baggage associate	<ul style="list-style-type: none"> ▶ Update access to regional baggage applications for their specific region. ▶ Read access to regional baggage applications for the other regions.
3200	Regional ground services catering associate	Update access to regional catering applications for their specific region.
3300	Regional ground services cleaning associate	Update access to regional cleaning applications for their specific region.
3400	Regional ground services airport liaison associate	Update access to regional airport liaison applications for their specific region.
4000	Customer services center manager	Approves requests for CSC applications for their specific CSC from personnel in their CSC.
4100	Customer service center associate	Update access to CSC applications for their specific CSC.

Employee code	Description	Requirements
5000	Regional IT services manager	Approve request for regional IT services and applications for their specific region from personnel in their region.
5100	Regional IT help desk associate	Password reset authority for all personnel in their specific region for Notes, Tivoli Access Manager, and z/OS accounts.
5110	Regional IT Linux associate	<ul style="list-style-type: none"> ▶ Total access to all Linux systems administration tools for that specific region. ▶ Password reset for Linux accounts for personnel working in their specific region. ▶ Grant access to Linux applications to personnel working in their specific region.
5120	Regional IT Windows Associate	<ul style="list-style-type: none"> ▶ Total access to all Windows systems administration tools for that specific region. ▶ Grant access to Windows applications to personnel working in their specific region.
5130	Regional z/OS associate	<ul style="list-style-type: none"> ▶ Total access to all z/OS systems administration tools for that specific region. ▶ Grant access to z/OS applications to personnel working in their specific region.
5140	Regional HR associate	Update access to the HR applications for that specific region.
6000	Core services executive	<ul style="list-style-type: none"> ▶ Approve all requests from the core services managers. ▶ Read access to all core services applications.
6100	Core services sales manager	<ul style="list-style-type: none"> ▶ Approve all requests from the core services sales associates. ▶ Update access to all core services sales applications.

Employee code	Description	Requirements
6110	Core services sales associate	Update access to all core services sales applications.
6200	Core services flight operations manager	<ul style="list-style-type: none"> ▶ Approve all requests from the core services flight operations associates and aircraft maintenance associates. ▶ Read access to all flight operations, crew, and aircraft maintenance applications.
6210	Core services flight operations associate	<ul style="list-style-type: none"> ▶ Update access to all flight operations and crew applications. ▶ Read access to all aircraft maintenance applications.
6220	Core services aircraft maintenance associate	<ul style="list-style-type: none"> ▶ Update access to all aircraft maintenance applications. ▶ Read access to all flight operations and crew applications.
6300	Core services support manager	<ul style="list-style-type: none"> ▶ Approve all core services human resource associate requests. ▶ Approve all core services accounting associate requests. ▶ Approve all core services IT Linux associate requests. ▶ Approve all core services IT Windows associate requests. ▶ Approve all core services z/OS associate requests. ▶ Approve all core services IT help desk associate requests. ▶ Update access to core services human resources applications. ▶ Update access to cross-reference accounting applications.
6310	Core services human resources associate	Update access to all core services human resources applications.
6320	Core services accounting associate	Update access to all core services accounting application.

Employee code	Description	Requirements
6410	Core services IT Linux associate	Total access to core services IT Linux tools.
6420	Core services IT Windows associate	Total access to core services IT Windows tools.
6430	Core services IT z/OS associate	Total access to core services IT z/OS tools.
6510	Core services IT help desk associate	<ul style="list-style-type: none"> ▶ Update access to all core services help desk tools. ▶ Password reset for any user core services user for Notes, Tivoli Access Manager, and z/OS accounts.

12.4.3 Roles

The combination of location and employee code are used to assign a functional role. See Table 12-5.

Table 12-5 TAA functional roles

Role	Description
Employee	Based on job code and location, access to all operational applications
Manager	Based on job code and location, access to all operational, financial, and human resources applications
Help desk	Based on job code and location, access to Tivoli Identity Manager services to reset passwords
Tivoli Identity Manager system administrator	Based on job code and location, access to all Tivoli Identity Manager configuration and reporting services
Executive	Based on job code and location, access to all operational, financial, and human resources applications
Aircraft maintenance	Based on job code and location, update access to all aircraft maintenance applications
Flight operations	Based on job code and location, update access to all flight operations applications
Flight operations manager	Based on job code and location, read only access to all aircraft maintenance and flight operations application

Additionally, a number of applicative roles, which are used to view and request accesses, also must be defined. See Table 12-6.

Table 12-6 TAA applicative roles

Role	Description
PC users	Employees assigned a PC by procurement services, based on job code
Mobile phone users	Employees assigned a PC by procurement services, manually requested

Definition rules are needed to assign a role to a person. The definition rules can be Lightweight Directory Access Protocol (LDAP) attributes that may be searched when a policy is executed. See Table 12-7.

Example for help desk role:

```
(&(objectclass=taaemployee)(|(employeetype=5100)(employeetype=6510)))
```

Table 12-7 TAA's Tivoli Identity Manager basic roles

Role	Definition rule
Employee role	All employee codes, except 6210 and 6220
Flight operations role	6210
Aircraft maintenance role	6220
Manager role	Employee codes 3000, 4000, 5000, 6100, and 6300
Executive role	1000, 2000, and 6000
Flight operations manager	6200
Help desk role	Employee codes 5100 and 6510
Tivoli Identity Manager administrators	Manually assigned
PC users	Employee codes 1000, 2000, 4000, 4100, 5000, 5100, 5110, 5120, 5130, and 5130
Mobile phone users	Manually requested

The roles shown in Table 12-7 on page 545 have been configured in Tivoli Identity Manager, as shown in Figure 12-17.

Select ^	Name ^	Description ^	Business Unit ^	Role Ty ^
<input type="checkbox"/>	<u>Aircraft Maintenance Role</u>	Aircraft Maintenance Role	<u>Tivoli Austin Airlines</u>	Static
<input type="checkbox"/>	<u>Employee Role</u>	Employee role, this is also a parent role.	<u>Tivoli Austin Airlines</u>	Static
<input type="checkbox"/>	<u>Executive Role</u>	Executive Role	<u>Tivoli Austin Airlines</u>	Static
<input type="checkbox"/>	<u>Flight Operations Manager</u>	Flight Operations Manager	<u>Tivoli Austin Airlines</u>	Static
<input type="checkbox"/>	<u>Flight Operations Role</u>	Flight Operations Role	<u>Tivoli Austin Airlines</u>	Static
<input type="checkbox"/>	<u>Help Desk Role</u>	Help Desk Role	<u>Tivoli Austin Airlines</u>	Dynamic
<input type="checkbox"/>	<u>ITIM Administrators</u>	Predefined system administrator role.	<u>Tivoli Austin Airlines</u>	Static
<input type="checkbox"/>	<u>Manager Role</u>	Manage role, this is also a parent role.	<u>Tivoli Austin Airlines</u>	Static
<input type="checkbox"/>	<u>Mobile Phone User</u>	Mobile Phone User	<u>Tivoli Austin Airlines</u>	Static
<input type="checkbox"/>	<u>PC User</u>	PC User	<u>Tivoli Austin Airlines</u>	Dynamic

Page 1 of 1 Total: 10 Displayed: 10 Selected: 0

Figure 12-17 TAA's organizational roles

Figure 12-18 shows the TAA roles and role hierarchy structures.

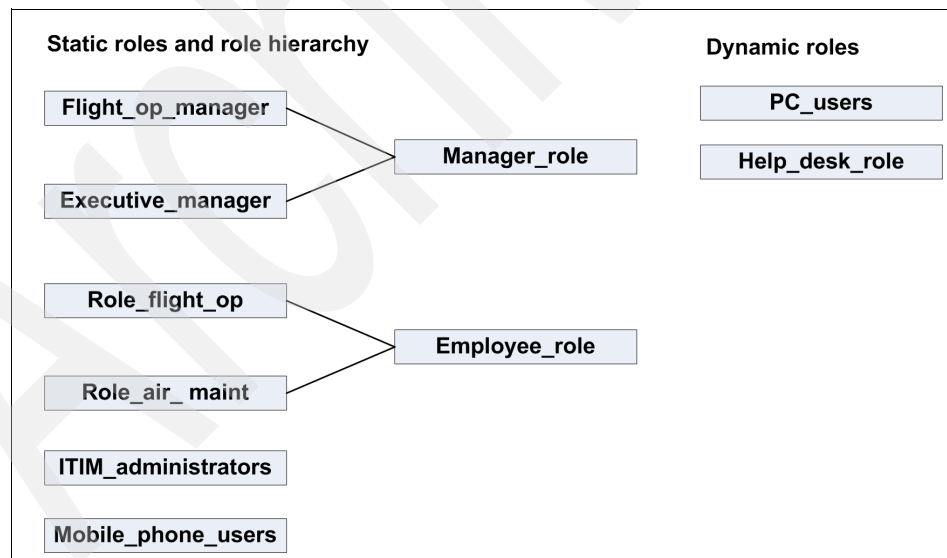


Figure 12-18 TAA organizational roles and role hierarchy structures

Once the roles are defined, you can build role hierarchy structures, as shown in Figure 12-18 on page 546, by linking static organizational roles together using parent → child relationships. Figure 12-19 shows an example of adding child roles to the manager role.

The screenshot displays the Tivoli Identity Manager web interface. On the left is a navigation menu with options like 'Home', 'Change Passwords', 'Manage Roles', 'Manage Organization Structure', 'Manage Users', 'Manage Services', 'Manage Groups', 'Manage Policies', 'Design Workflows', 'Set System Security', 'Reports', 'Configure System', 'View Requests', 'Manage Activities', 'About', and 'Log Out'. The main area is titled 'Manage Roles' and contains a table of roles. A context menu is open over the 'Manager Role' row.

Roles
You can add, change, delete roles, manage role membership, or manage role hierarchy. Select the role in the table, and then click the appropriate button or use the context menu for a specific role.

10 results found for: *

Select	Name	Description	Business Unit	Role
<input type="checkbox"/>	Aircraft Maintenance Role	Aircraft Maintenance Role	Tivoli Austin Airlines	Static
<input type="checkbox"/>	Employee Role	Employee role, this is also a parent role.	Tivoli Austin Airlines	Static
<input type="checkbox"/>	Executive Role	Executive Role	Tivoli Austin Airlines	Static
<input type="checkbox"/>	Flight Operations Manager	Flight Operations Manager	Tivoli Austin Airlines	Static
<input type="checkbox"/>	Flight Operations Role	Flight Operations Role	Tivoli Austin Airlines	Static
<input type="checkbox"/>	Help Desk Role	Help Desk Role	Tivoli Austin Airlines	Dynai
<input type="checkbox"/>	ITIM Administrators	Predefined system administrator role.	Tivoli Austin Airlines	Static
<input type="checkbox"/>	Manager Role	Parent role.	Tivoli Austin Airlines	Static
<input type="checkbox"/>	Mobile Phone User		Tivoli Austin Airlines	Static
<input type="checkbox"/>	PC User		Tivoli Austin Airlines	Dynai

Page 1 of 1 Total: []

Close

Figure 12-19 Manage Role: Add Child Role

Role approvals

TAA will be delegating some role approval processes directly to the TAA managers who are making the approval decision that historically the Identity Manager administrators have implemented. This automates part of the business that was previously slow and labor intensive. The users can request access to these roles byway of the self-care interface. An approval work item will then be sent to the role owner for approval or rejection. If approved, automatic provisioning policies will then set up the required entitlements on the managed resources. Initially, this will be limited to the static organizational roles shown in Table 12-8.

Table 12-8 Organizational roles requiring role approval

Role name	Role owner ^a
Aircraft maintenance role	Flight operations manager
Flight operations role	Flight operations manager

a. Roles can be owned by persons or other roles.

Tivoli Identity Manager workflow extensions are used to define role approvals. These are performed by editing person entity operation workflows and using these extensions to configure the approval processes. A number of extensions have been provided to enable various types of role approvals to be performed. These examples can be found in the following directory:

```
<ITIM install directory>\extensions\examples\workflow\roleApproval
```

Using the administrative interface, go to **Configure System** → **Manage Operations**. Following the example for *simple role owner approval*, update the entity level operational workflows shown in Figure 12-20 on page 549.

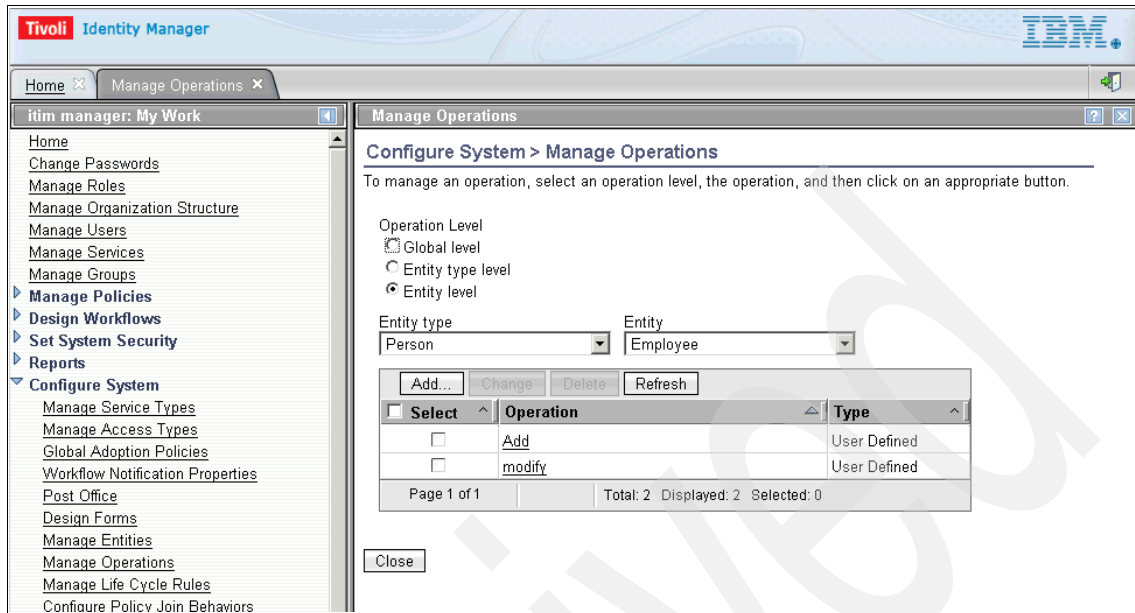


Figure 12-20 Manage operations

Note: Role management tasks can also be scripted using APIScript. For more details and an example, see Appendix C, “Automating tasks for role management” on page 643.

Role approval: additional TAA customization

TAA will not define owners for all roles initially; this will be implemented in stages, starting with the roles previously defined in Table 12-8 on page 548. This requires that additional customization be completed so that only roles with defined owners are passed to the extension *approveRoleByOwner* by the person add or modify operation (Figure 12-20).

Access the operational workflow, shown in Figure 12-20 on page 549, and add an additional script node and transitions to check for the existence of roles and role owners before passing the workflow execution to the *approveRoleByOwner* extension. Figure 12-21 shows the completed workflow.

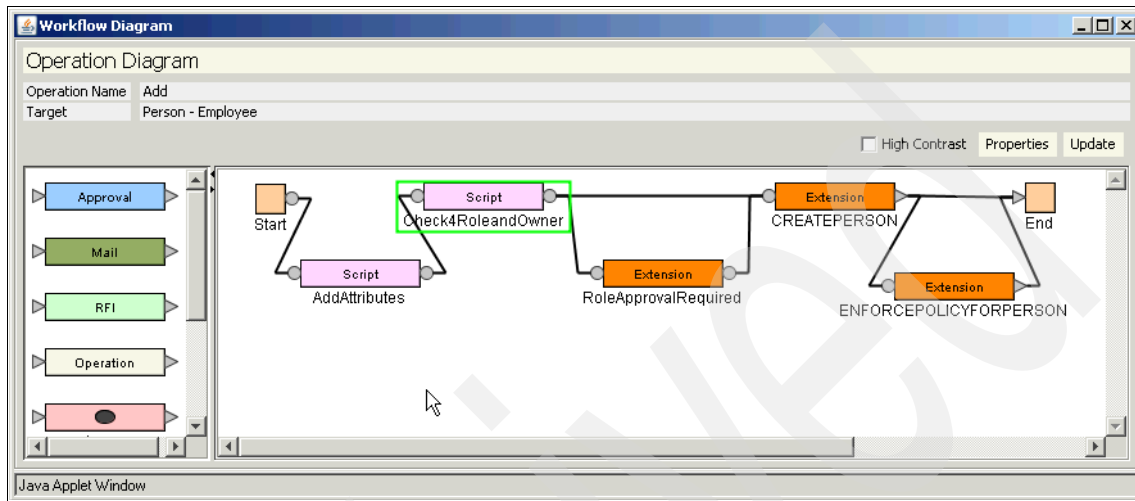


Figure 12-21 Add entity person: *taaemployee* customization

The following information enables you to create the script node and transition lines required to complete the customization update shown in Figure 12-21.

Note: The script node *AddAttributes* was added as part of the HR feed and is not related to the customization described in this section.

Perform these steps:

1. Open the *add* person operation and, using the workflow editor, click **Properties** and define a new relevant data string called *roleOwnerExists*. See Figure 12-22 on page 551.

Properties

Operation Type Static Non Static

Input Parameters Add Modify Delete

ID	Type
container	OrganizationalContainer
person	Person

S: Subject R: Requestee B: Both

Output Parameters Map Relevant Data Add Modify Delete

ID	Type	Relevant Data ID
----	------	------------------

Relevant Data Add Modify Delete

ID	Type
roleOwnerExists	String

S: Subject R: Requestee B: Both

Ok Cancel

Java Applet Window

Figure 12-22 workflow properties: relevant data

- Now add a script node and join the transition lines, as shown in Figure 12-21 on page 550. The script node should contain the JavaScript shown in Example 12-2.

Example 12-2 Script node JavaScript: check4RoleandOwner

```

roleOwnerExists.set("false");
var per = person.get();
var rolesArray = per.getNewRoles();
if (rolesArray != null)
{
    for( var i=0; i<rolesArray.length;i++) {
        var owners = rolesArray[i].getProperty("owner");
        if(owners.length!=0){
            roleOwnerExists.set("true");
            break;
        }
    }
}

```

- The last step is to update the transition lines to contain the correct condition details. Figure 12-23 shows the transition that joins the script node *Check4RoleandOwner* to the node *RoleApprovalRequired*, which contains the extension *approveRoleByOwner*. This path is used when roles, which contain a role owner, are found.

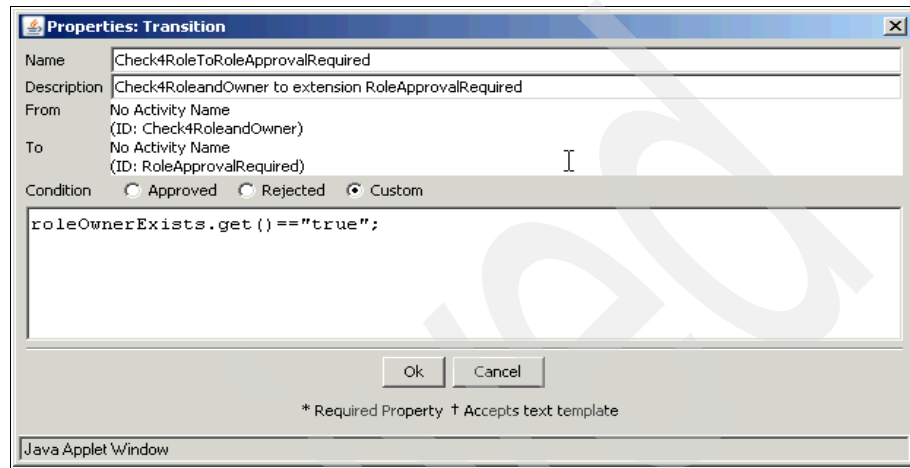


Figure 12-23 Workflow transition: for roles and role owner found

- Figure 12-24 shows the transition for *check4RoleandOwner* to *CREATEPERSON*, which is used when no roles are found or the role does not have an role owner defined.

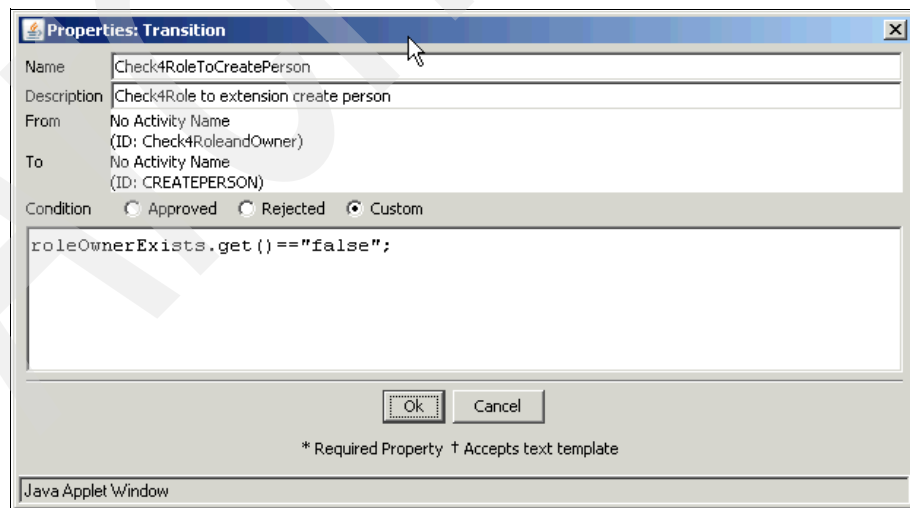


Figure 12-24 Workflow transition: no roles or role owner found

This completes the additional customization on the *add* person operational workflow. Now repeat the process to update the *modify* operational workflow.

Delegating role management

The Tivoli Identity Manager base configuration does not provide a means for role owners to manage the roles for which they are responsible directly through the administration interface.

TAA requires that all role owners who have delegated role management responsibilities be able to manage their organization roles using the administrative interface. This requires additional Identity Manager customization. The following summarizes the configuration steps that are required to provide this customization:

- ▶ Define role management view (select **System Security** → **Manage Views**).
- ▶ Create a Tivoli Identity Manager group for the role owners.
 - Assign the newly created role management view to this group.
 - Add the role owners to the group as members.
- ▶ Create an ACI to allow role owners update permission.
 - Select **Protection Category** → **Static Organizational Role**.
 - Select **Operation** → **Grant All**.
 - Select **Permissions** → **Grant All**.
 - Select **Membership** → *The owner of the role*.

Define a view for role management

Using the administrative interface, select **Set System Security** → **Manage Views**. Click **Create** and create the view *role management*, as shown in Figure 12-25 and Figure 12-26 on page 555. When you are done, click **OK**.

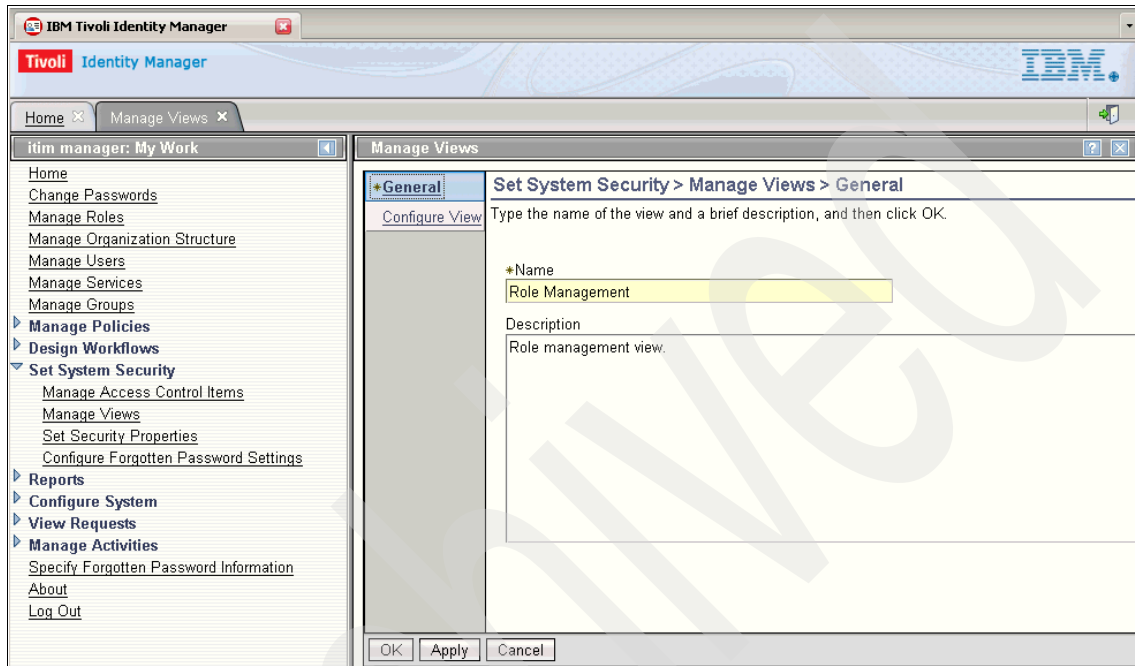


Figure 12-25 Manage view: General

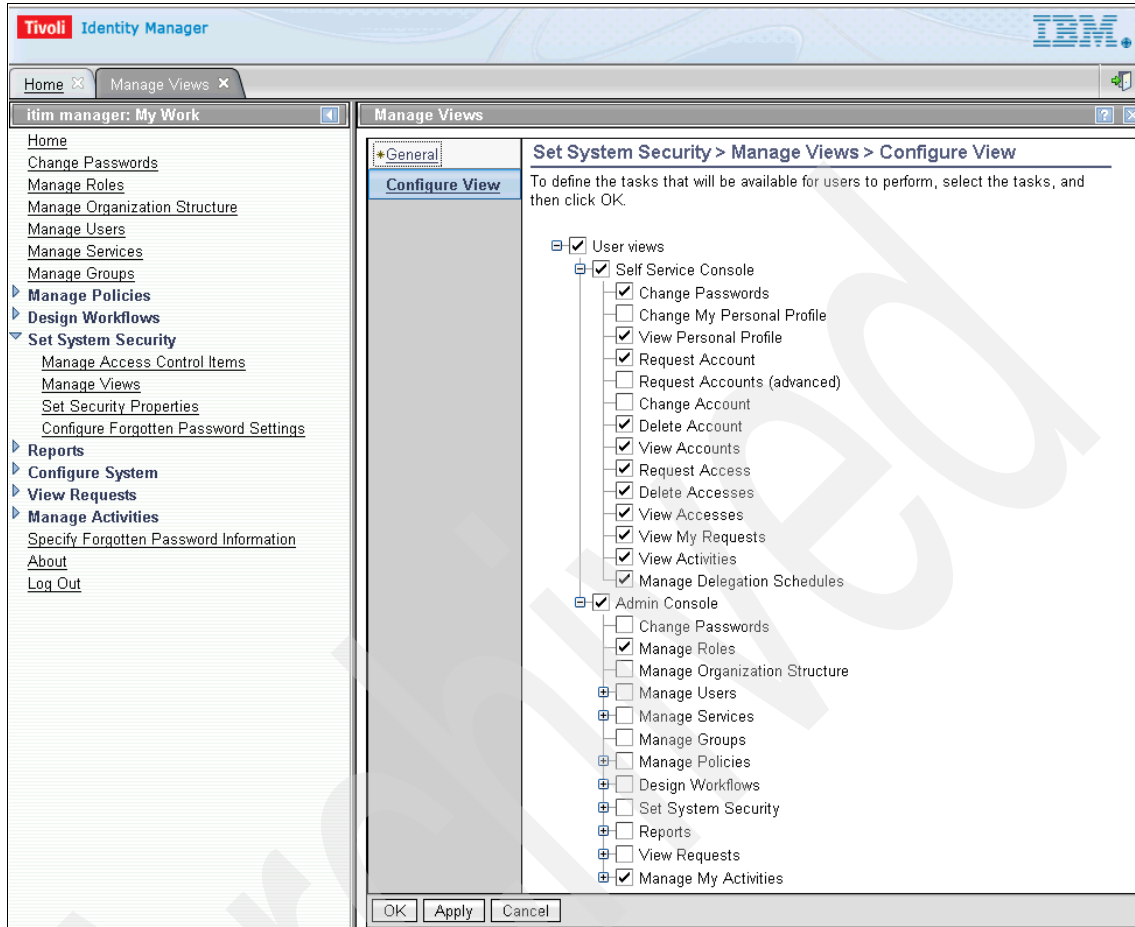


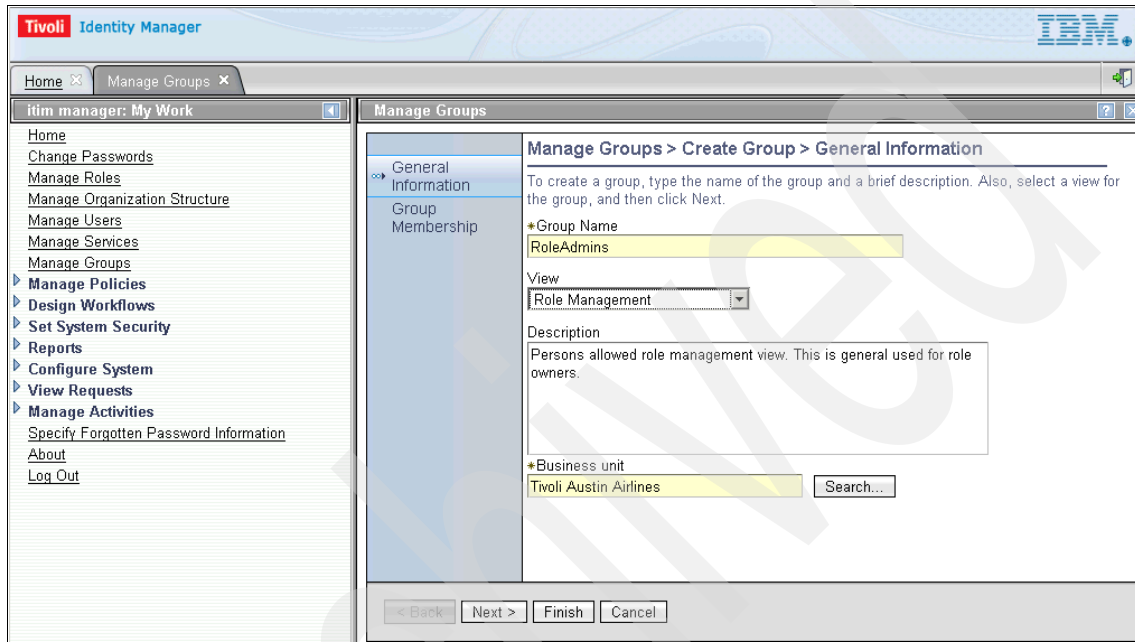
Figure 12-26 Manage view: Configure View

Define an Tivoli Identity Manager group for role administration

Now that we have defined a view, we can create a group for role administration using the administration interface Manage Group function.

Perform these steps:

1. Using the administration interface, select **Manage Group**, then search for the service ITIM. Click **Create**, enter the details to create a new group called *RoleAdmins*, and assign the newly defined view *role management*. Figure 12-27 shows the completed form.



The screenshot shows the Tivoli Identity Manager administration interface. The main window is titled 'Manage Groups' and displays the 'Create Group > General Information' form. The left sidebar contains a navigation menu with options like 'Home', 'Change Passwords', 'Manage Roles', 'Manage Organization Structure', 'Manage Users', 'Manage Services', 'Manage Groups', 'Manage Policies', 'Design Workflows', 'Set System Security', 'Reports', 'Configure System', 'View Requests', 'Manage Activities', 'Specify Forgotten Password Information', 'About', and 'Log Out'. The main content area has a breadcrumb trail 'Manage Groups > Create Group > General Information' and instructions: 'To create a group, type the name of the group and a brief description. Also, select a view for the group, and then click Next.' The form fields are: '*Group Name' with the value 'RoleAdmins', 'View' with a dropdown menu set to 'Role Management', 'Description' with the text 'Persons allowed role management view. This is general used for role owners.', and '*Business unit' with the value 'Tivoli Austin Airlines' and a 'Search...' button. At the bottom, there are navigation buttons: 'Back', 'Next >', 'Finish', and 'Cancel'.

Figure 12-27 Manage group: Create Group General tab

2. Click **Next** to proceed to the next configuration window, shown in Figure 12-28 on page 557. Using this Group Membership window, you may add users who will be allowed to use the role management view. These users should be roles owners, who require update permissions on the roles for which they are responsible.

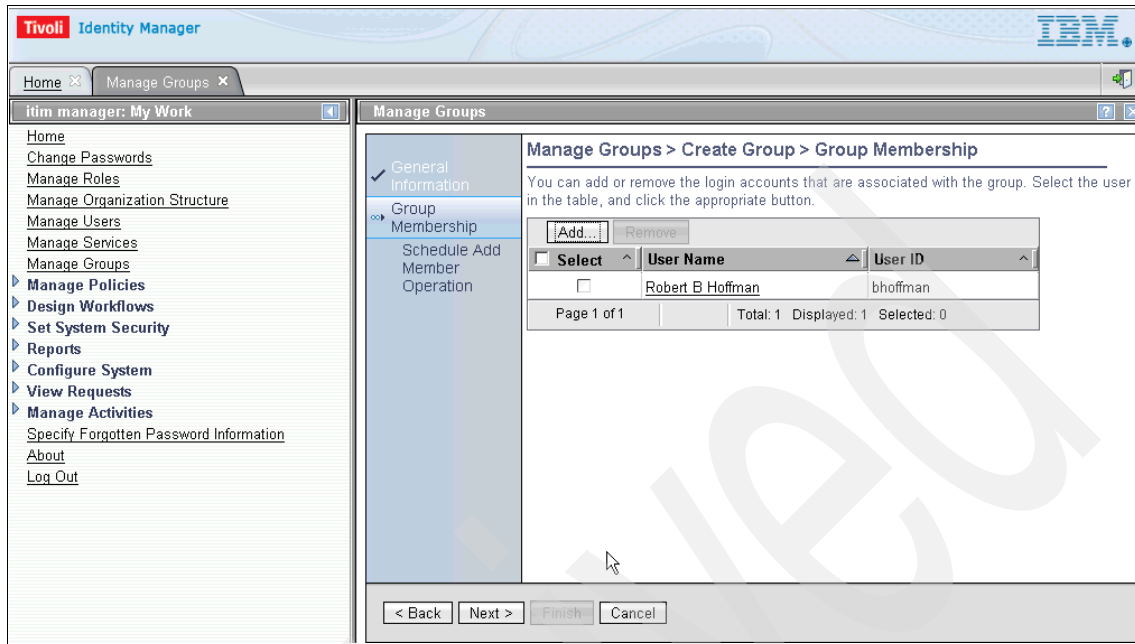


Figure 12-28 Manage group: add group membership

Define ACI for role administration

The final step is to create the ACI to grant the role owner permission to update and manage the roles for which they are responsible.

Perform these steps:

1. Complete the General ACI window, as shown in Figure 12-29. Be sure to select the protection category of **Static Organizational Role**.

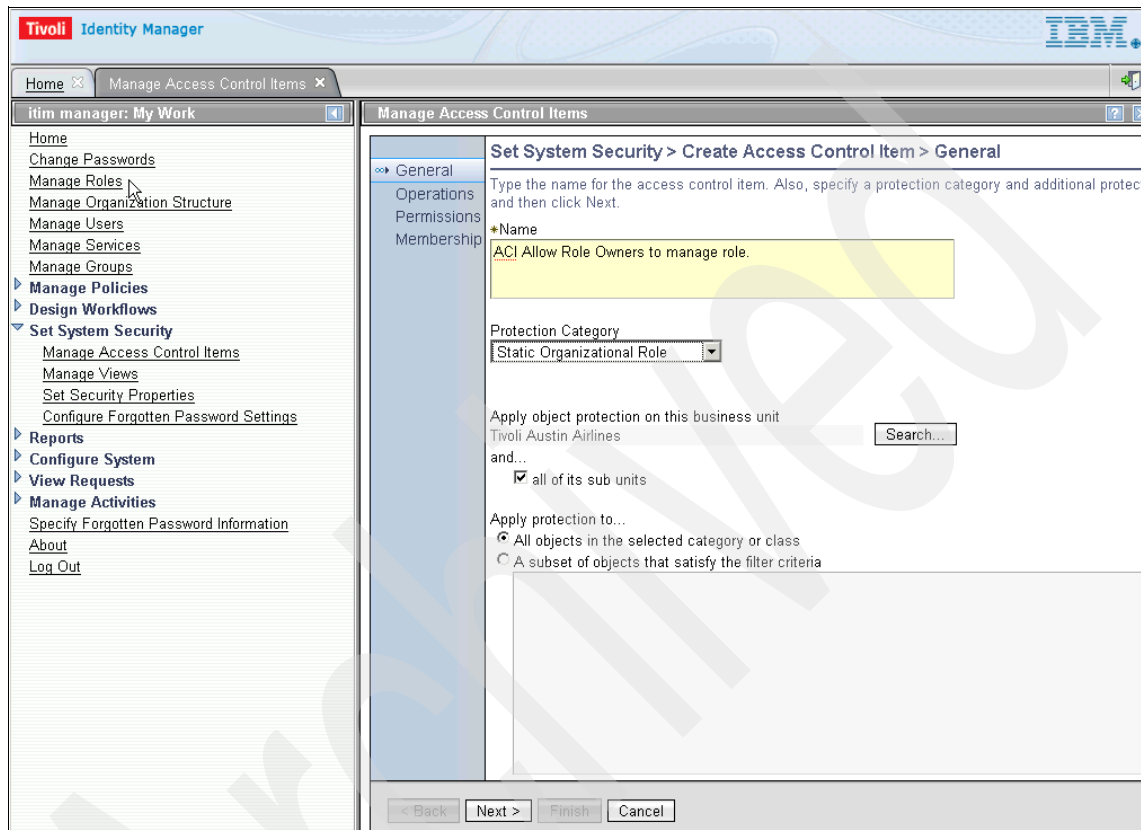


Figure 12-29 Manage access control items: General tab

2. Click **Next** to proceed to the Operations window. See Figure 12-30.

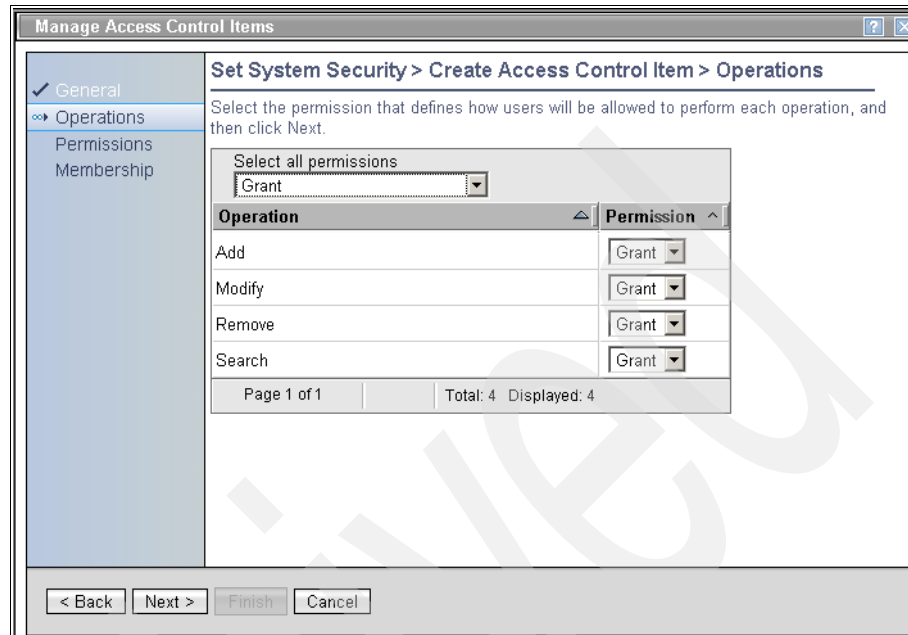


Figure 12-30 Manage access control items: Operations tab

3. Click **Next** and update the Permissions window, as shown in Figure 12-31.

Manage Access Control Items

Set System Security > Create Access Control Item > Permissions

Select the permission that defines how users will be allowed to perform each attribute, and then click Next.

Select all read: Grant | Select all write: Grant

Attribute	Read	Write
Access Options	Grant	Grant
Child roles	Grant	Grant
Classification	Grant	Grant
Description	Grant	Grant
Name	Grant	Grant
Others	Grant	Grant
Owner	Grant	Grant

Page 1 of 1 | Total: 7 Displayed: 7

< Back | Next > | Finish | Cancel

Figure 12-31 Manage access control items: Permissions tab

4. Click **Next** and complete the Membership window. Check the box **The owners of the role** and then click **Finish** to submit the new ACI, as shown in Figure 12-32.

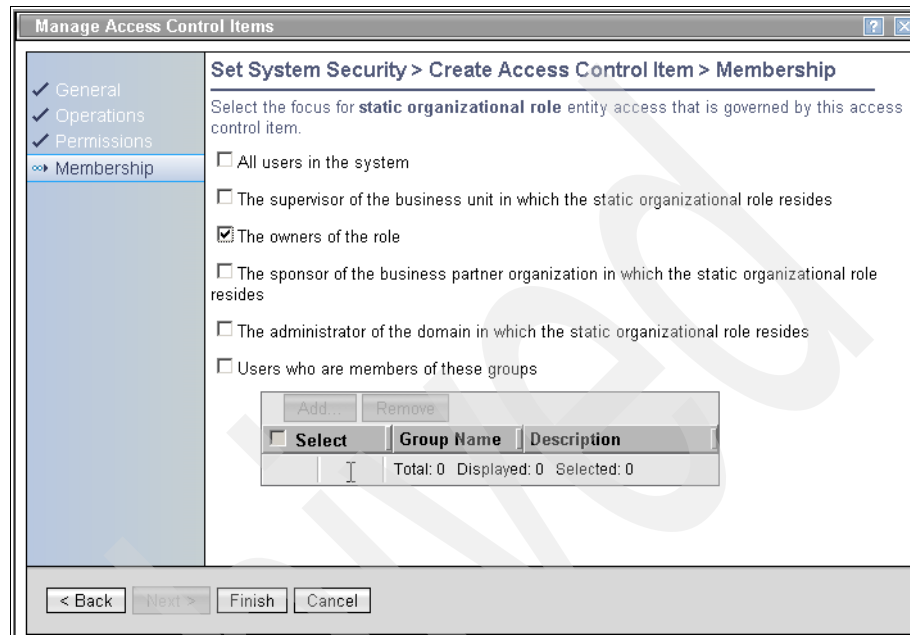


Figure 12-32 Manage access control items: Membership tab

Using role administration customization

Users defined as members of the ITIM group *RoleAdmins* are now able to access the administration interface. The Manage Role function now appears in their view. The newly defined ACI *allow role owners to manage role* allows role administrators to manage the roles for which they are responsible, as shown in Figure 12-33.

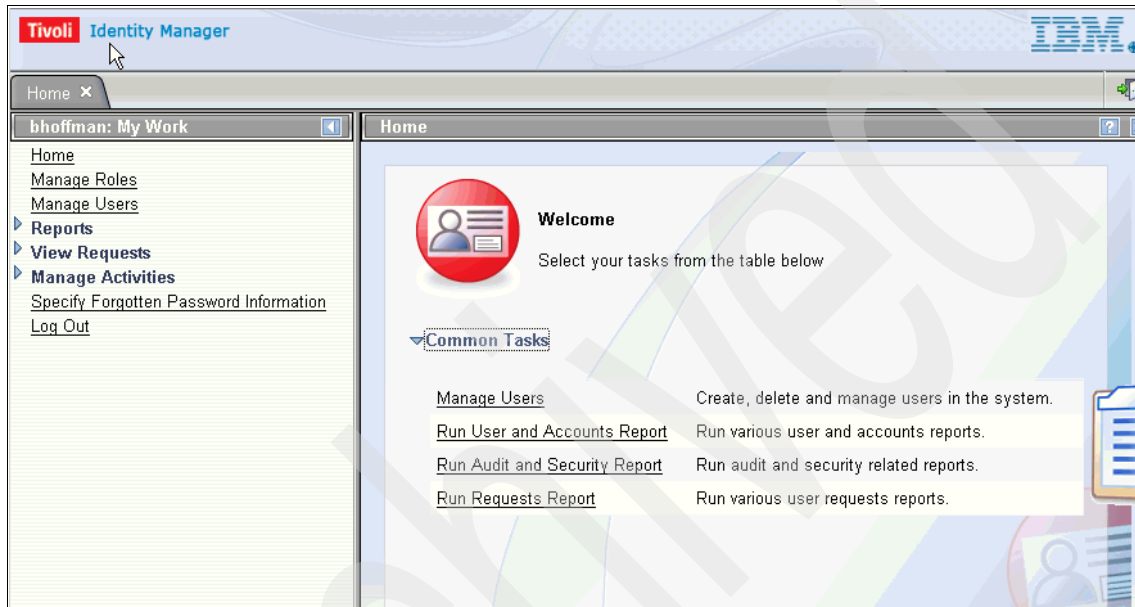


Figure 12-33 Administrative interface: role administrator view

12.4.4 Provisioning policies

The combination of job code and location determines the Tivoli Identity Manager organizational roles. The organizational role then determines the entitlement for a managed service and executes an associated provisioning policy. A service selection policy may also be used to determine access to a specific managed service instance.

One technique is to develop broad provisioning policies and use service selection policies within them to grant access to specific managed resources within them. This technique may provide the greatest coverage of users while reducing the number of policies, thereby avoiding conflicting and overlapping definitions. See Table 12-9 on page 563 for an overview of TAA's provisioning policies.

Table 12-9 TAA phase IV provisioning policies

Provisioning policy	Membership	Entitlement
employee	All	<ul style="list-style-type: none"> ▶ Tivoli Identity Manager at the central data center ▶ Tivoli Access Manager at the central data center ▶ Lotus Notes at the central data center
central_employee	role_employee	<ul style="list-style-type: none"> ▶ RACF at the central data center (employee group) ▶ Tivoli Access Manager (central employee group) ▶ Tivoli Access Manager at the central data center (read_flight_op group) ▶ Tivoli Access Manager at the central data center (read_air_maint group)
local_employee	role_employee	<p>Use service selection policies to use the location and job code to determine access to managed resources located near the employee:</p> <ul style="list-style-type: none"> ▶ Nearest Windows Active Director Domain (employee group) ▶ Nearest Linux servers (employee group)
central_manager	role_manager	<ul style="list-style-type: none"> ▶ RACF at the central data center (manager group) ▶ Tivoli Access Manager at the central data center (manager group) ▶ Tivoli Access Manager at the central data center (read_flight_op group) ▶ Tivoli Access Manager at the central data center (read_air_maint group)
local_manager	role_manager	<p>Use service selection policies to use the location and job code to determine access to managed resources located near the employee:</p> <ul style="list-style-type: none"> ▶ Nearest Windows Active Director Domain (manager group) ▶ Nearest Linux servers (manager group)
help_desk	role_help_desk	Tivoli Identity Manager at the central data center (help_desk group)

Provisioning policy	Membership	Entitlement
itim_manager	role_itim_manager	Tivoli Identity Manager at the central data center (itim_manager group)
central_flight_op	role_flight_op	▶ Tivoli Access Manager at the central data center (update_flight_op group)
central_air_maint	role_air_maint	▶ Tivoli Access Manager at the central data center (update_air_maint group)

As discussed in 12.1.3, “Defining roles” on page 521, the use of provisioning policies and organizational roles can entitle a person with the correct access on different systems. Using this feature, combined with the discussion about service selection policies in “Service selection policies” on page 422, TAA *has* implemented RBAC.

TAA has created the provisioning policies (as shown in Table 12-9 on page 563) and assigned the correct entitlements and organizational roles as required. Next, Figure 12-34 shows the entitlement details for TAA's managers for its Tivoli Access Manager accounts.

The screenshot shows the Tivoli Identity Manager console. The left sidebar contains a navigation tree with 'Manage Policies' expanded. The main content area is titled 'Manage Provisioning Policies > Entitlement Parameter'. Below the title is a table of entitlement parameters. The table has columns for 'Select', 'Name', 'Template value', 'Enforcement', and 'Value Type'. The parameters listed are:

Select	Name	Template value	Enforcement	Value Type
<input type="checkbox"/>	Description	Account created by Tivoli Identity Manager	Default	Constant Value
<input type="checkbox"/>	Full name	return subject.getProperty("cn")[0];	Default	JavaScript
<input type="checkbox"/>	Single Signon Capability	true	Default	Constant Value
<input type="checkbox"/>	Distinguished Name	return "cn="+subject.getProperty("cn");	Default	JavaScript
<input type="checkbox"/>	Last name	return subject.getProperty("sn")[0];	Default	JavaScript
<input type="checkbox"/>	Group Membership	manager	Default	Constant Value

At the bottom of the table, it shows 'Page 1 of 1', 'Total: 6', 'Displayed: 6', and 'Selected: 0'. There are 'Continue' and 'Cancel' buttons below the table.

Figure 12-34 Entitlements for managers

It assigns the Tivoli Access Manager group *managers* to the users in the organizational role *role_manager*, because the membership for this provisioning policy is *role_manager*, as shown in Figure 12-35.

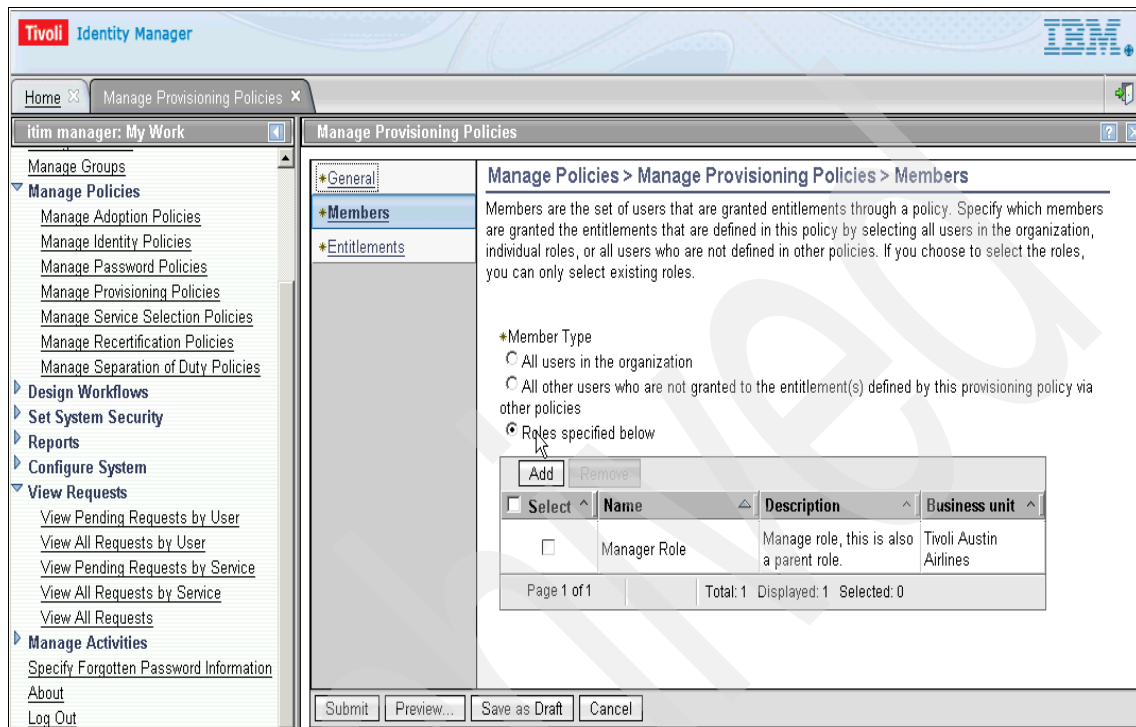


Figure 12-35 Provisioning policy membership

As discussed in 12.1.4, “Defining provisioning policies” on page 528, the attribute for groups shown in Figure 12-34 on page 565 has been set to mandatory.

12.4.5 Separation of duty policy

During their data gathering, TAA identified a conflict of interest between the roles *air maintenance*, *flight operations*, and *flight operations manager*. TAA requires that only members of flight operations should have update access to flight operations applications and data. Likewise, only members of air maintenance should have update access to air maintenance applications and data. All members of the roles central employee and central manager are granted read access only.

Membership of roles flight operations and air maintenance must be approved using role approval by the role flight operations manager. This was previously

defined in “Role approvals” on page 548. The separation of duty policy will ensure that you may only be a member of one of the following roles at any one time:

- ▶ Flight operations manager
- ▶ Flight operations
- ▶ Air maintenance

Any exemptions to the separation of duty (SoD) policy must be approved by the policy owner, which will be defined as the role *executive manager*. By default, all SoD policy approval work items are routed to the policy owner. A graphical representation of the TAA SoD policies is depicted in Figure 12-36.

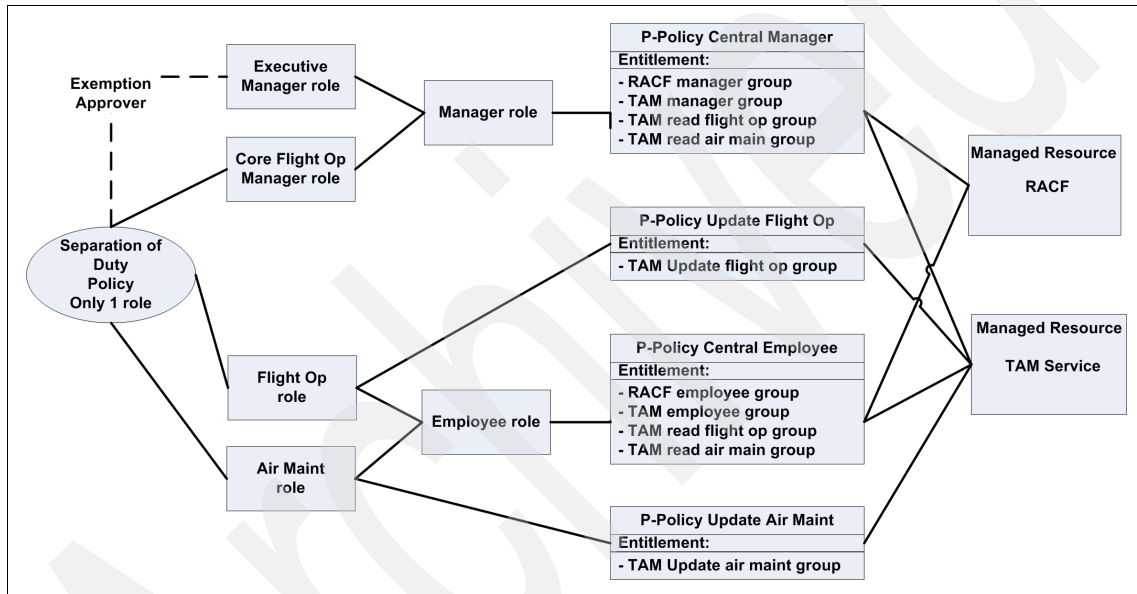


Figure 12-36 TAA separation of duty overview

Defining separation of duty policy

In Table 12-10, we identify the details of the SoD policy to be defined.

Table 12-10 TAA SoD policy details

Policy name	Target roles	Allowed	Owner
Separate flight operations and air maintenance updates	<ul style="list-style-type: none"> ▶ Flight operations ▶ Air maintenance ▶ Flight operations manager 	One	Executive manager

Let us now walk through the setup process for the SoD policies:

1. Using the administrative interface, select **Manage Policies** → **Manage Separation of Duty Policies** → **Create Separation of Duty Policy**. See Figure 12-37.

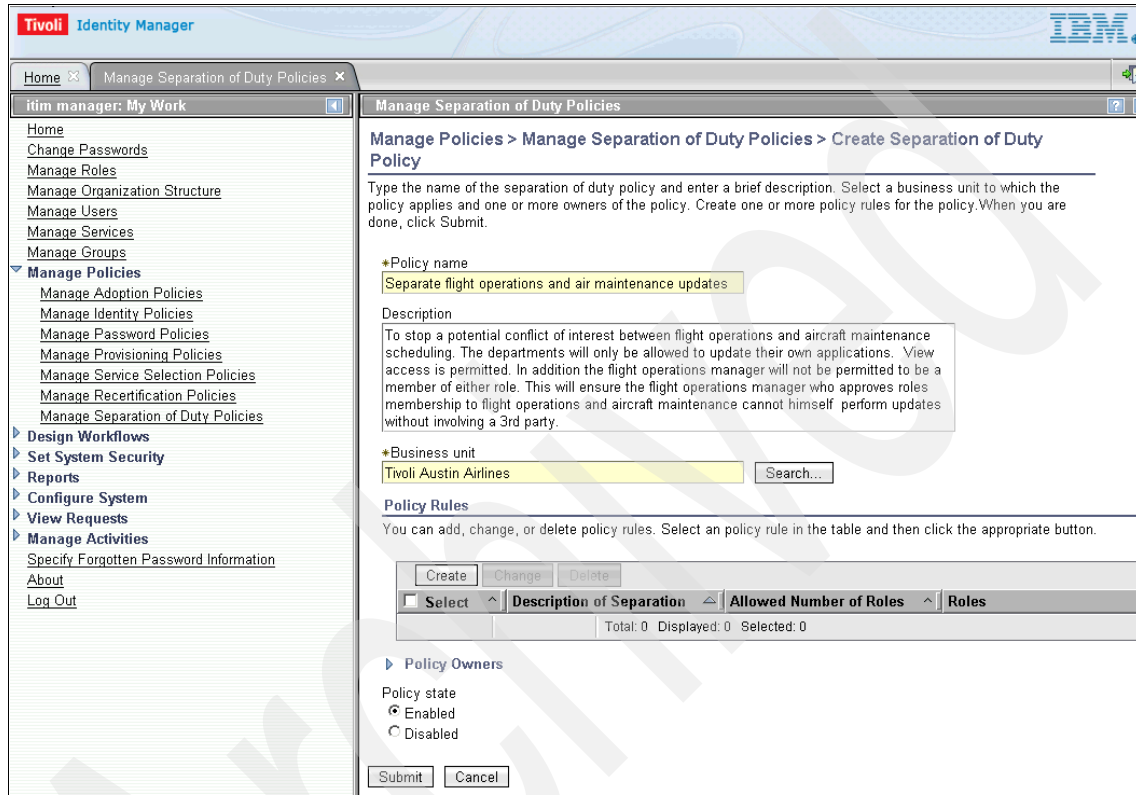


Figure 12-37 TAA Create Separation of Duty Policy window

Enter the policy name, description, and owner for the policy. The description is used in any notification messages sent with regard to this policy. The policy owner receives all requests for exemptions to the policy. Once you have entered the basic policy details, create the policy rules by clicking **Create**.

- Provide a policy name for the rule, then add the roles in the scope of this policy. Finally, define the allowed number of roles. This is the number of roles in the rule that one person maybe a member of at one time without causing a policy violation. Figure 12-38 shows a completed form.

The screenshot shows the 'Create Policy Rule' interface in Tivoli Identity Manager. The breadcrumb path is 'Manage Policies > Manage Separation of Duty Policies > Create Policy Rule'. The form includes a description field with the text 'Aircraft maintenance and flight operations role separation.' and a 'Build Role Separation List' section. This section contains a search bar, a 'Quick Add' form with a 'Role name' input and an 'Add' button, and a table of roles. The table has columns for 'Select', 'Name', 'Description', and 'Business Unit'. Three roles are listed: 'Aircraft Maintenance Role', 'Flight Operations Manager', and 'Flight Operations Role'. At the bottom, there is a dropdown for 'Allowed number of roles' set to '1', and 'OK' and 'Cancel' buttons.

Manage Policies > Manage Separation of Duty Policies > Create Policy Rule

Type a name for the policy rule and add two or more role names to the role list. Select the number of roles in the role list a user can belong to, and click OK to apply the rule to the policy.

+Description of separation
Aircraft maintenance and flight operations role separation.

Build Role Separation List
Specify a list of static roles that will restrict membership to the allowed number. Type the name of the role below and click Add to add it to the role list. Click Search to search and add roles on a separate screen.

Search...

Quick Add
Role name
Add

Select	Name	Description	Business Unit
<input type="checkbox"/>	Aircraft Maintenance Role	Aircraft Maintenance Role	Tivoli Austin Airlines
<input type="checkbox"/>	Flight Operations Manager	Flight Operations Manager	Tivoli Austin Airlines
<input type="checkbox"/>	Flight Operations Role	Flight Operations Role	Tivoli Austin Airlines

Page 1 of 1 Total: 3 Displayed: 3 Selected: 0

Allowed number of roles
1

OK Cancel

Figure 12-38 Create separation of duty policy rules

- Once the policy rules have been defined, click **OK** to return to the create policy view, as shown in Figure 12-39. Then submit the policy to complete the create process by clicking **Submit**.

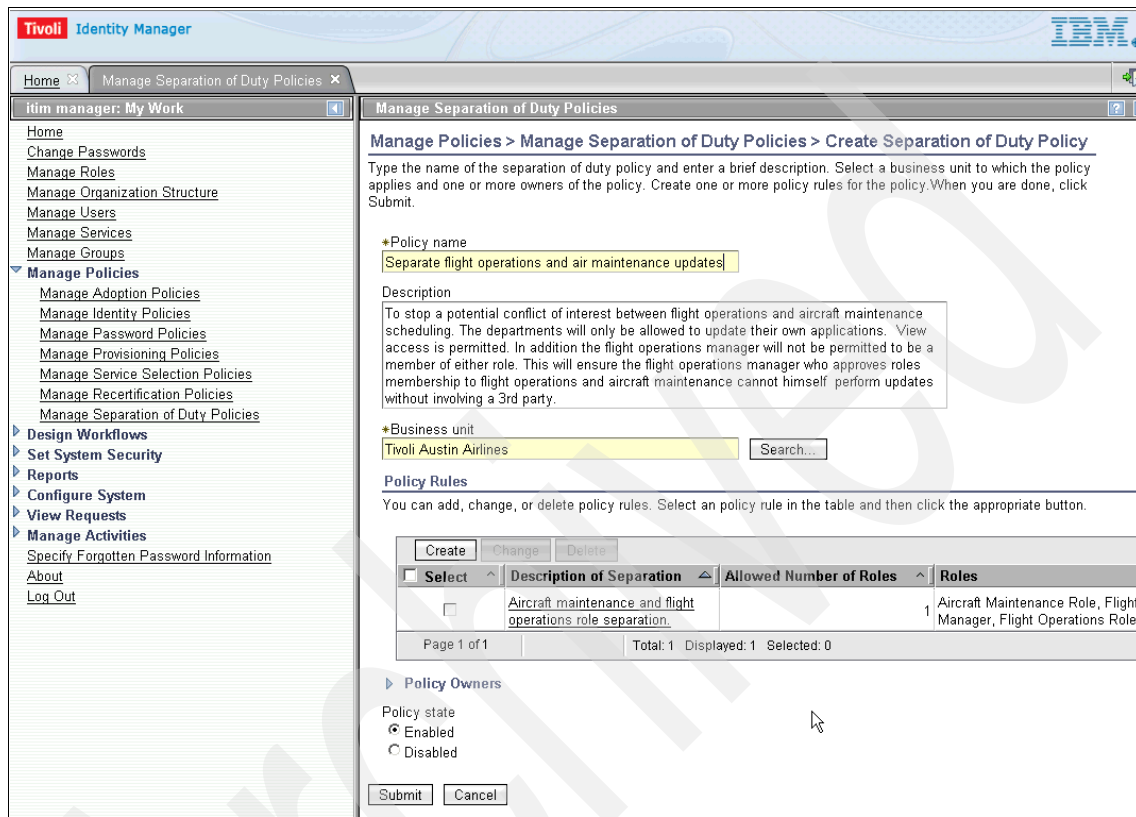


Figure 12-39 TAA separation of duty policy

Evaluating separation of duty policies

Once you have defined the separation of duty policy, you should carry out an evaluation. This allows you to review any violations. Any violations should be rectified or exemptions granted.

Perform these steps:

1. Select **Manage Policies** → **Manage Separation of Duty Policies**. Select the policy you just created, as shown in Figure 12-39 on page 570, and then evaluate the policy by clicking the **Evaluate** button, as shown in Figure 12-40.

The screenshot shows the Tivoli Identity Manager web interface. The left sidebar contains a navigation menu with options like Home, Change Passwords, Manage Roles, Manage Organization Structure, Manage Users, Manage Services, Manage Groups, Manage Policies (expanded), Design Workflows, Set System Security, Reports, Configure System, View Requests, Manage Activities, Specify Forgotten Password Information, About, and Log Out. The main content area is titled 'Manage Separation of Duty Policies' and includes a search bar with a search button. Below the search bar, there is a section titled 'Separation of Duty Policies' with a sub-header '1 results found for: *'. A table of policies is displayed with columns: Select, Policy Name, Description, Business Unit, State, Violations, and Exemptions. The table contains one row with a checked 'Select' checkbox, the policy name 'Separate flight operations and air maintenance updates', a detailed description, the business unit 'Tivoli Austin Airlines', the state 'Enabled', 1 violation, and 0 exemptions. At the bottom of the table, it shows 'Page 1 of 1' and 'Total: 1 Displayed: 1 Selected: 1'.

Select	Policy Name	Description	Business Unit	State	Violations	Exemptions
<input checked="" type="checkbox"/>	Separate flight operations and air maintenance updates	To stop a potential conflict of interest between flight operations and aircraft maintenance scheduling. The departments will only be allowed to update their own applications. View access is permitted. In addition the flight operations manager will not be permitted to be a member of either role. This will ensure the flight operations manager who approves roles membership to flight operations and aircraft maintenance cannot himself perform updates without involving a 3rd party.	Tivoli Austin Airlines	Enabled	1	0

Figure 12-40 TAA evaluate separation of duty policy

- The evaluation detected one policy violation. To view the details, click the number in the Violations column. Figure 12-41 shows the violation in more detail. From here an administrator or policy owner can approve or revoke exemptions to this violation. To correct the violation, remove the person identified in the violation from one of the roles.

The screenshot shows the Tivoli Identity Manager interface. The left sidebar contains a navigation menu with categories like 'Home', 'Manage Policies', 'Design Workflows', 'Set System Security', 'Reports', 'Configure System', 'View Requests', and 'Management Activities'. The main content area is titled 'Manage Separation of Duty Policies' and displays a summary for the policy 'Separate flight operations and air maintenance updates'. It shows 1 violation and 0 exemptions. A table lists the violation details for user Robert B Hoffman on September 18, 2009, with conflicting roles 'Aircraft Maintenance Role, Flight Operations Manager'.

Manage Policies > Manage Separation of Duty Policies > Violations and Exemptions Summary

The following are summaries of violations and exemptions by policy rule for the policy **Separate flight operations and air maintenance updates**. Click on a specific policy rule name to see details about the violations and exemptions for that policy rule.

Total number of violations: **1**
 Total number of exemptions: **0**

Order rules
 By violation

Aircraft maintenance and flight operations role separation. ✖ 1 Violations ✔ 0 Exemptions

1 Violations for Rule Aircraft maintenance and flight operations role separation.

Select	Date of Violation	User Name	User Roles in Conflict	Policy Roles in Conflict
<input type="checkbox"/>	September 18, 2009 1:52:18 PM	Robert B Hoffman	Aircraft Maintenance Role, Flight Operations Manager	Aircraft Maintenance Role, Flight Operations Manager

Page 1 of 1 Total: 1 Displayed: 1 Selected: 0

0 Exemptions for Rule Aircraft maintenance and flight operations role separation.

Select	User Name	Approve	Date Approv	User Roles in Confl	Policy Roles in Confl	Approval Notes
Total: 0 Displayed: 0 Selected: 0						

Figure 12-41 TAA separation of duty: display violation

Delegating separation of duty policy management

The Tivoli Identity Manager base configuration does not provide a means for the separation of duty policy owners to manage the policy directly through the administration interface. By default, the policy owner interacts with the policy byway of the exemption approval process. For example, when a policy violation is triggered, the policy owner receives an approval work item. This can either be approved to create an exemption or rejected to leave the configuration with a policy violation. If the approval work item is approved, the exemption will be created; if the approval work item is rejected, the requested role will *not* be added (request canceled, without role membership change).

TAA requires that all separation of duty policy owners are able to review and manage the policies for which they are responsible by way of the administration interface. This requires additional Identity Manager customization. The following configuration steps are required to provide this customization;

- ▶ Define separation of duty policy view (select **System Security** → **Manage Views**).
- ▶ Create an ITIM group for the separation of duty policy owners.
 - Assign the newly created separation of duty policy view to the group.
 - Add the separation of duty policy owners to the group as members.
- ▶ Create an ACI to grant policy owners update permission.
 - Select Protection Category → Separation of Duty Policy.
 - Select Operation → Grant all.
 - Select Membership → The owner of the policy.

12.4.6 Accesses

In order to allow users to request access to resources from the Tivoli Identity Manager self-care interface, as well as to provide a view to which accesses have already been requested of a user, Tivoli Identity Manager accesses can be defined.

Defining access types

Access types can be defined to categorize accesses. Tivoli Austin Airlines requires accesses that fall into a set of categories including procurement and application administration groups. Therefore these access types will be defined in Tivoli Identity Manager.

Table 12-11 Tivoli Austin Airlines additional access types

Access type key	Access type description
ApplAdminGroup	Application administration group
Procurement	Procurement group for requesting resources

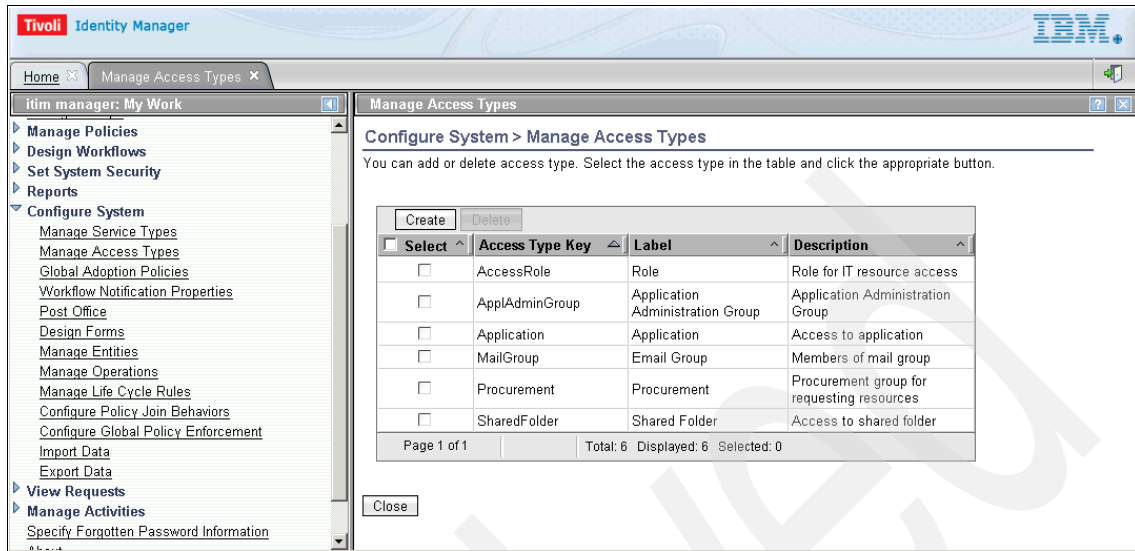


Figure 12-42 Tivoli Austin Airlines access types configuration

Figure 12-42 shows how access types are configured in Tivoli Identity Manager's system configuration menu. Labels are configured by editing the CustomLabels.properties files in the Tivoli Identity Manager server data directory. Here the custom labels file has been edited with the following extract appended at the end of the file:

```
# CUSTOM LABELS - Access types
ApplAdminGroup=Application Administration Group
Procurement=Procurement
```

Note that if the Tivoli Identity Manager language pack has been installed, regional versions of this customization must be propagated to internationalized custom label files.

Defining role accesses

Role accesses are defined in the main role configuration window. Figure 12-43 shows the TAA mobile phone user role definition parameters. Notice that at the bottom of the window the three options available to optionally define the role as an access show the role in the common access list and access type selection. These options determine how users will be able to request the access.

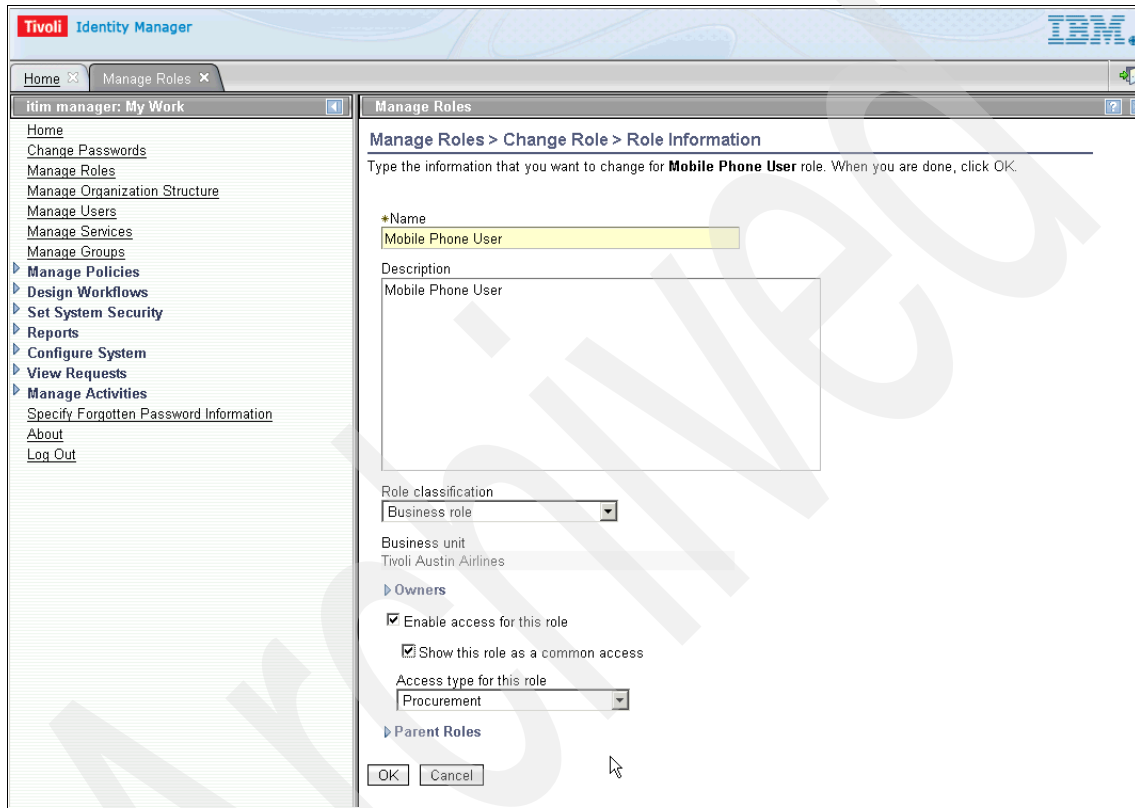


Figure 12-43 TAA mobile phone user role and corresponding access definition

TAA will expose three organizational roles to common role access. Following the example shown in Figure 12-43 on page 575, we define the following roles to have access, as shown in Table 12-12.

Table 12-12 TAA role and access definitions

Role name	Access type
Mobile phone user	Procurement
Aircraft maintenance role	Role
Flight operations role	Role

Defining group accesses

Group accesses can be managed by using the Manage Groups access view. Perform these steps:

1. Search for and select the service for which you wish to define a group access. Then search for the group on which you are to define the access. Figure 12-44 shows the group *AirMaint* on the Active Directory service *MS.AD.AustinCSC*. Note that the access status is currently disabled.

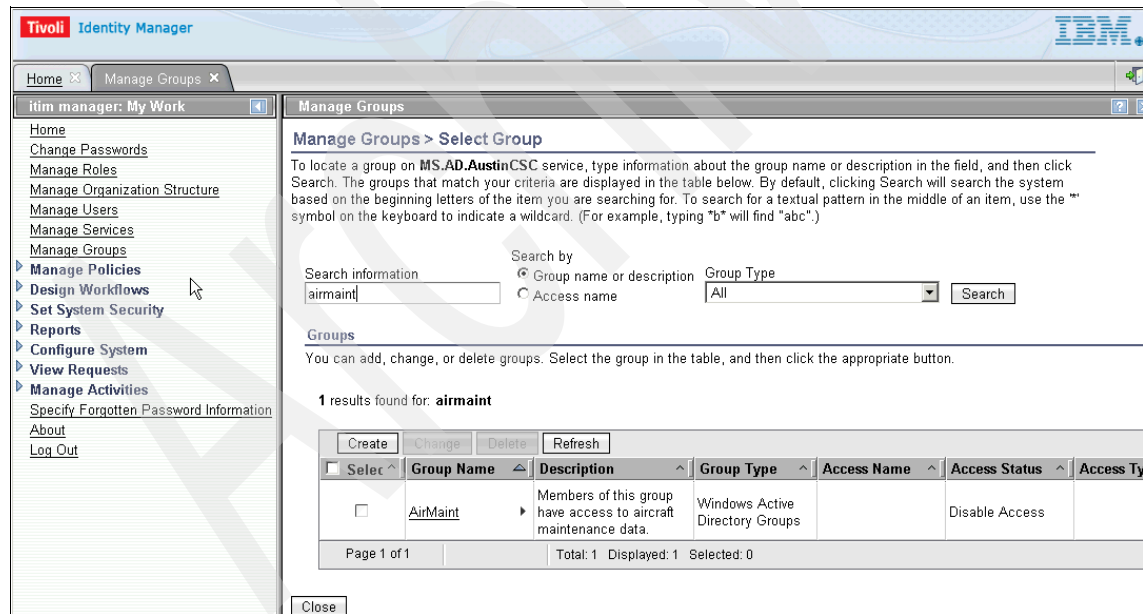


Figure 12-44 TAA Group Management Active Directory service

2. To define a group access, select the group and click the **Change** button, and then select the **Access Information** tab. Figure 12-45 shows the completed group access form.

The screenshot shows the Tivoli Identity Manager interface. The left sidebar contains a navigation menu with options like Home, Change Passwords, Manage Roles, Manage Organization Structure, Manage Users, Manage Services, Manage Groups, Manage Policies, Design Workflows, Set System Security, Reports, Configure System, View Requests, Manage Activities, Specify Forgotten Password Information, About, and Log Out. The main content area is titled 'Manage Groups > Change Group > Access Information'. It features a 'General Information' tab and an active '*Access Information' tab. The form includes a checkbox for 'Define an Access' (checked), 'Access status' options (Enable Access, Enable Common Access, Disable Access), an 'Access name' field with the value 'Aircraft Maintenance Role', an 'Access type' dropdown set to 'Application', an 'Access description' text area containing 'The role controls who has update access to aircraft maintenance applications.', an 'Access owner' field with 'Search...' and 'Clear' buttons, an 'Approval workflow' dropdown set to 'Management Approval', and two checked checkboxes: 'Notify users when access is provisioned and available for use' and 'Notify users when access is de-provisioned'. 'OK' and 'Cancel' buttons are at the bottom.

Figure 12-45 TAA Group Management: define group access

3. Repeat the process to define any additional group accesses.

Figure 12-44 on page 576 and Figure 12-45 on page 577 have shown how service groups and accesses are mapped and how they can be configured to appear in the Common Access menu available to self-care users. Figure 12-46 shows a TAA user using the self-care interface to view the common accesses that he has access to as a result of the configuration performed to the roles and service group accesses.

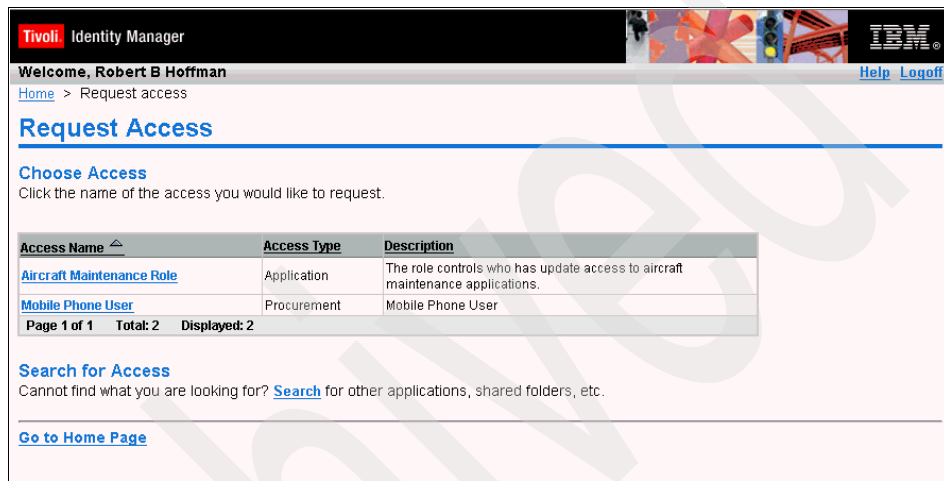


Figure 12-46 TAA self-care user common accesses request view

12.5 Certification process

Recertification of the continuing need for access to accounts, groups, and roles is part of any good identity life cycle management process. Tivoli Identity Manager provides this functionality as part of the product.

12.5.1 Requirements

TAA has decided to implement a monthly certification process for accounts on the most critical systems. In TAA's case, Linux has been identified as being a critical system within the scope of this phase. As discussed in 8.2, "Functional requirements" on page 325, temporary access can easily result in forgotten access privileges. This may become a particular problem if those accounts are no longer required by a person. As a result, the need for having an account has to be checked every month. The owner of the resource initially must certify the need for this access. This is then followed by the person's supervisor certifying the need for this access.

In addition to the monthly recertification of critical system access, TAA implements an annual recertification of all role and group memberships for all employees. The employee's manager is responsible for the recertification. The recertification takes place in January of each year and the managers will have 14 days to complete the recertification, after which the recertification will be automatically approved.

12.5.2 Design considerations

Recertification policies give Tivoli Identity Manager users with appropriate rights the ability to design revalidation sequences of events that are triggered based on time intervals, specific dates, or immediately.

Recertification policies can be set for a user type, for example, person or business partner person, accounts, or access entities. A single policy may apply to multiple entities. Entities of type account and access can only have one policy applied to them.

12.5.3 TAA's critical account recertification implementation

All Linux user account holders must have their need for an account periodically revalidated. In order to accomplish this, a recertification policy executes at 8 a.m. on the second day of each month to allow the owner of the relevant Linux accounts to recertify their continuing need for the account. This is followed by the request being passed through to the person's supervisor for approval. For TAA the Linux monthly recertification policy is created for this purpose.

General configuration of the policy

A new recertification policy is created with the name Linux monthly recertification policy and an appropriate description.

1. Navigate to **Manage Policies** → **Manage Recertification Policies**. Click the **Create** button to begin your configuration. The first panel is the General tab as shown in Figure 12-47. Enter the information as shown and click **Next**.

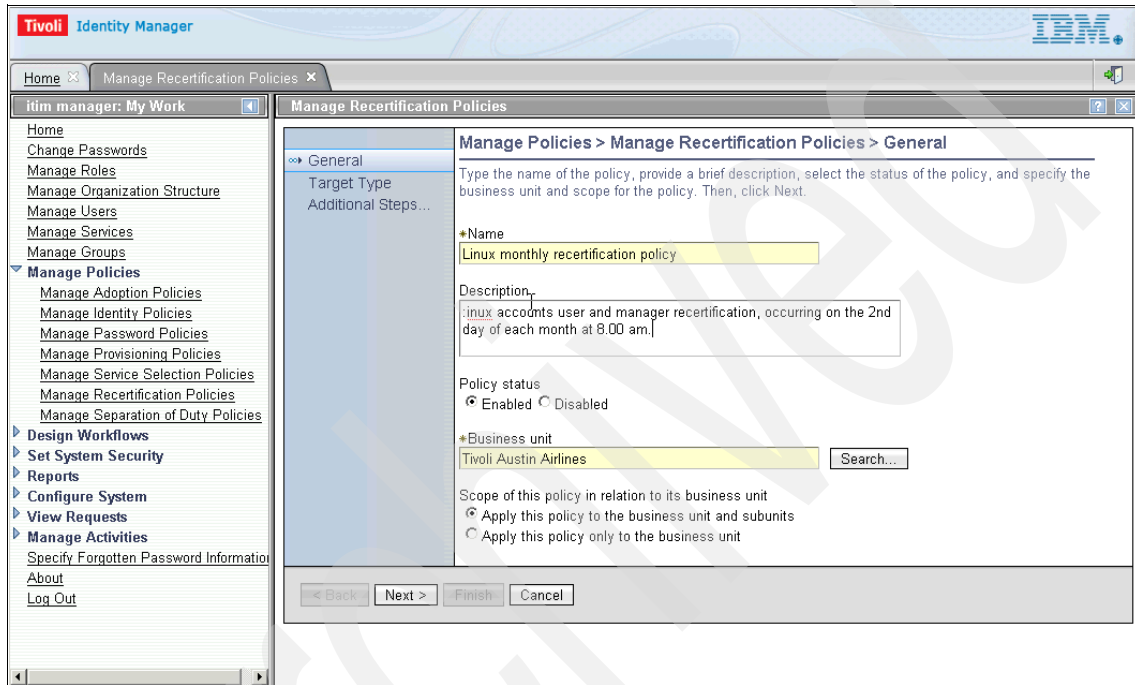


Figure 12-47 General properties of the new Linux monthly recertification policy

2. In the Target Type panel you can define whether the recertification applies to accounts, accesses or users. In this case we want this policy to apply to the accounts themselves, as reflected in Figure 12-48. Click **Next** to continue.

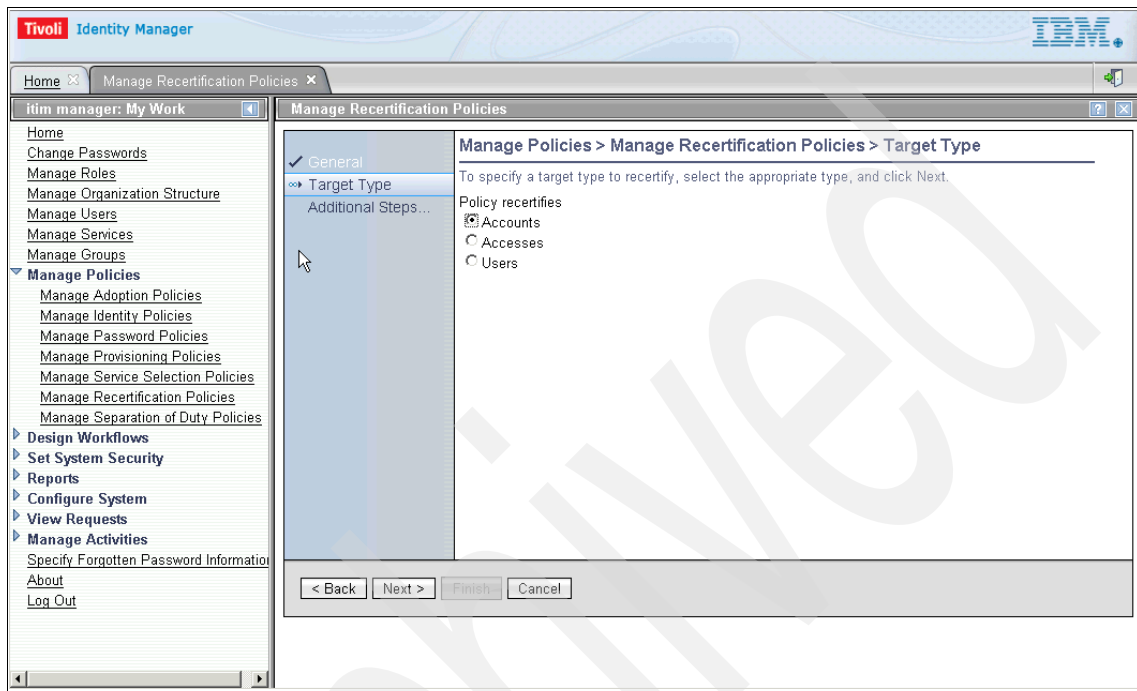


Figure 12-48 Recertification policy target type selection tab

3. In the next panel, Service Target, you select the actual policy targets. In this case the policy should apply to the TAA Linux service. You need to click the **Add** button in order to select the service from a list of all available services. Then the TAA Linux service appears in the Service Target list as shown in Figure 12-49. Click **Next** to continue.

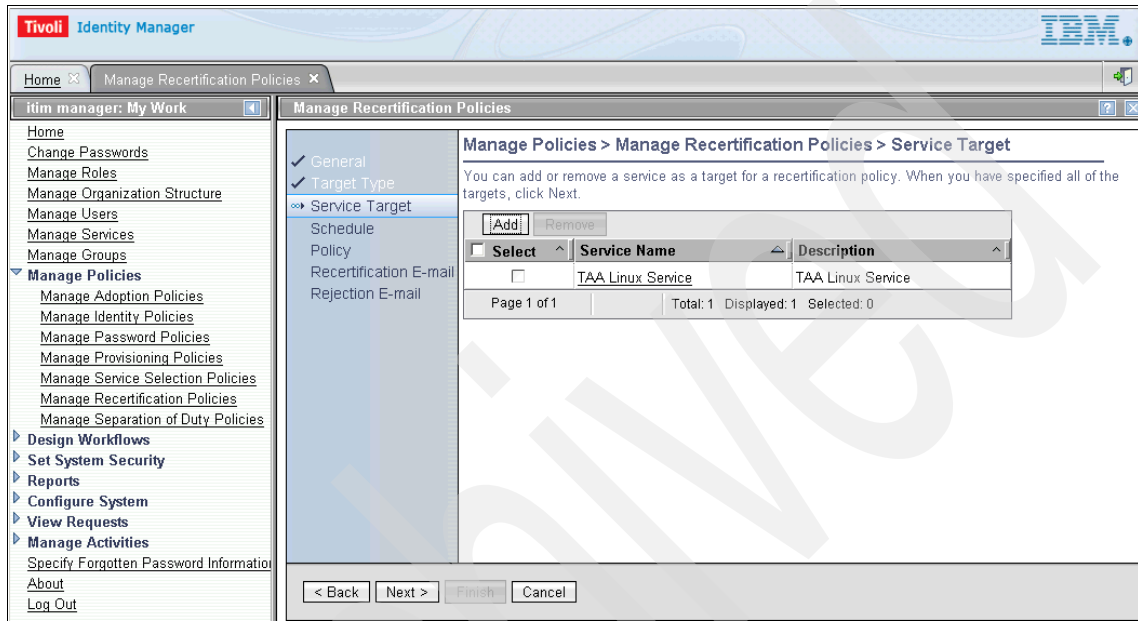


Figure 12-49 Recertification policy service selection tab

4. In the next panel, Schedule, you determine the scheduling of the policy. In this instance, it is scheduled to occur on a calendar basis, every second day of the month at 8:00 a.m. Figure 12-50 reflects this requirement. Click **Next** to continue.

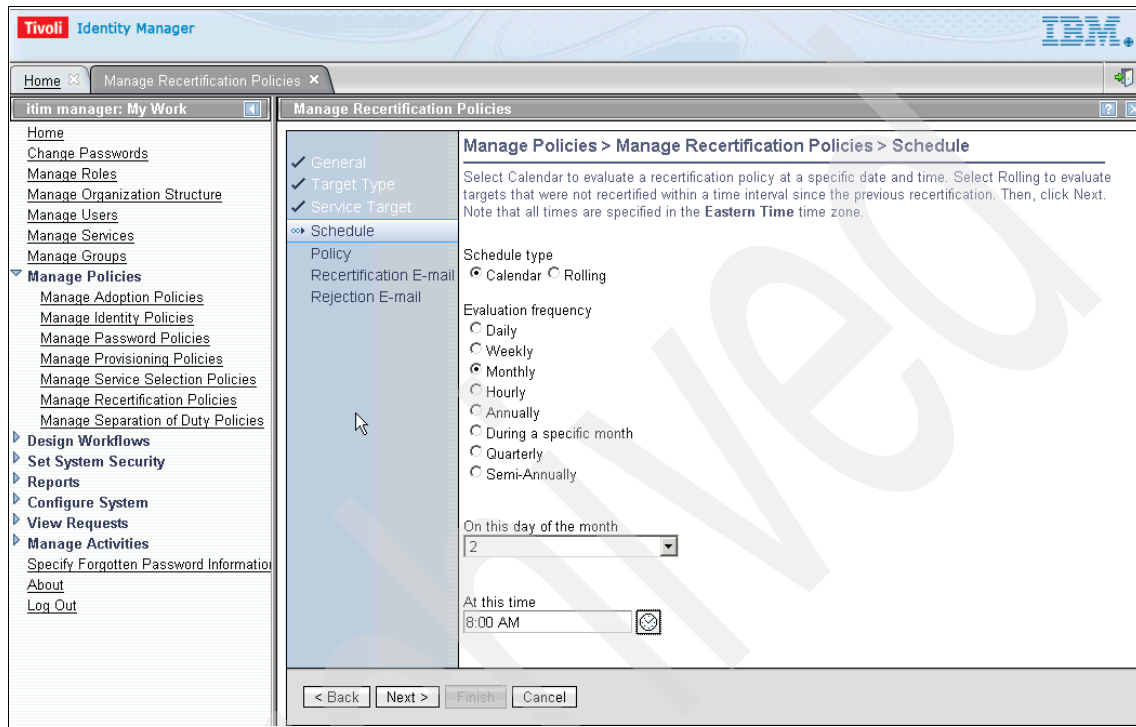


Figure 12-50 Recertification policy schedule tab

5. The next step involves the creation of the policy's workflow. Two options are presented at this point:
- A simple workflow wizard
 - An advanced workflow design graphical interface

The simple workflow wizard only allows for single-level approval, so the advanced workflow design graphical interface is required to create a workflow satisfying TAA's requirements. Select **Advanced** and the workflow wizard starts automatically. The workflow interface, as well as the corresponding workflow for TAA, is shown in Figure 12-51.

Note: For better overview and a resizable workflow application, you may want to use the function *Launch as separate window*.

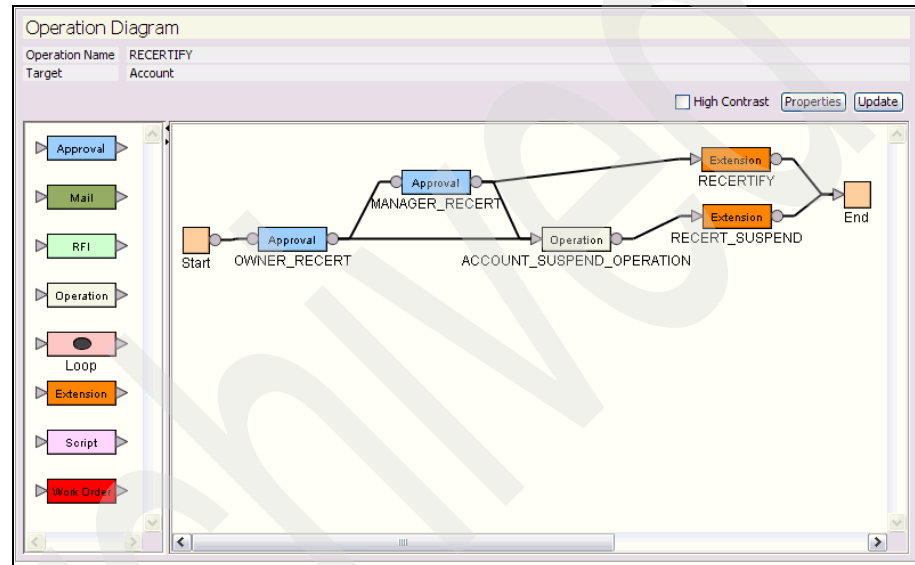


Figure 12-51 Linux recertification policy advance workflow

The workflow depicted here shows the final design of the workflow. A first recertification approval node has been configured to request a recertification approval from the account owner. If approved within the configured time frame, the process flow is redirected toward the second approval node. This second approval node is set up for manager approval. If the manager also approves the account recertification within the configured time frame, then the account will be recertified. If any of the two participants reject the recertification or let the request time out, the account will be suspended.

The workflow's approval nodes should be configured as follows.

- a. The first approval in the flow, immediately after the start node, is the owner approval node. By double-clicking it, or right-clicking and selecting **Properties**, its properties can be defined.

Figure 12-52 shows the general properties of the owner approval node. The activity ID must be entered without spaces, and is displayed in the workflow, as well as in transition-line properties. In this case it is called OWNER_RECERT.

ID	Type	Relevant Data ID
entity	Account	Entity
service	Service	theService
owner	Person	Owner

Figure 12-52 Owner approval general properties

The activity name has been entered as \$ITIM_RECERTIFY. The dollar sign (\$) in front indicates that this is a label, which in the Tivoli Identity Manager interfaces is translated into a meaningful name in the different languages configured. In the default English language this label is configured to translate as *Recertification Approval*. This label, as well as other workflow labels, can be found in the `Labels.properties` files in the Tivoli Identity Manager *data* directory. Additional labels can be configured by editing the `CustomLabels.properties` files in the Tivoli Identity Manager *data* directory. However, they may not appear in reports.

The participant of the approval has been set to be the requestee, that is, the person for whom the process was executed. No escalation participant has been defined in the requirements, and therefore none has been configured here. Escalation participants are participants to whom the activity is routed to if the main participant cannot be identified or has not actioned the activity in the allocated time.

The escalation limit has been configured to one week, or seven days. Participants (and escalation participants if configured) will have the amount of time defined in this field to approve or reject this approval request.

The join and split types are both set to *AND*. These directives define how the workflow node uses transition lines:

- The join type is a workflow directive that synchronizes incoming transitions.

The AND condition forces the activity to wait for all active incoming transition (information flow) paths to be completed before initiating the activity.

The OR condition allows the activity to initiate after one of the activity's transition condition evaluates as true and complete.

- The split type is a workflow directive that synchronizes diverging transitions.

The AND condition traverses all transitions whose information flow condition evaluates to true.

The OR condition traverses only the first transition that evaluates to true.

The entity type should be set to *Account*. This configures the approval node for account-related approvals.

Finally, the input parameters for the approval must be defined, as they will be used to identify what the approval is for. The *Relevant Data ID* column is empty when the approval node is first created, and requires to be mapped to the corresponding workflow properties of the same type (these can be configured by clicking the **Properties** button at the top of the workflow advanced design interface). By clicking one of these parameters and then on **Search Relevant Data**, a sub-window opens that allows you to select the appropriate property.

- b. The next task is to configure the notification properties of the owner approval node, as shown in Figure 12-53 on page 587. These should already be configured in the default workflow that appears in the workflow advanced interface, and should be kept as is.

General Notification Action Text Postscript

Use Default Template

† Subject

```
<RE key="recertTemplateSubject"><PARM><JS>process.subject</JS></PARM><P
```

Text XHTML

† Message Body

```
<RE key="recertTemplateBody"><PARM><JS>process.subject</JS></PARM><
<RE key="recertDeclineSuspendsBody"><PARM><JS>process.subject</JS>
<RE key="name"/>: <RE><KEY><JS>process.name;</JS></KEY></RE>
```

Use Group Email Topic

Group Email Topic

OWNER_RECERT

Ok Cancel

* Required Property † Accepts text template

Figure 12-53 Owner approval notification properties

The Use Group Email Topic check box should be checked, and the field below should contain a value of OWNER_RECERT (the same as the name of the node). Group email topics are used by Tivoli Identity Manager's post office, which can be configured to aggregate notification mails with the same topic together over a defined period of time before sending them. This reduces the number of notifications that a user may receive over time.

c. Next, we proceed to the Action Text tab, as depicted in Figure 12-54.

The screenshot shows a configuration window with four tabs: General, Notification, Action Text (selected), and Postscript. The 'Action Text' tab contains the following fields:

- * Approval Code: AA
- * Rejection Code: AR
- Cue Text: RECERTAPPROVAL
- † Action Text: (empty)

Below the 'Action Text' field, there is a checked checkbox labeled 'Use Notification Text as Action Text'.

At the bottom of the window, there are 'Ok' and 'Cancel' buttons, and a legend: '* Required Property † Accepts text template'.

Figure 12-54 Owner approval action text properties

The approval and rejection codes should be set respectively to AA and AR. These code are used by the workflow engine to process the activity's results, and by Tivoli Identity Manager to display the result of the activity. They stand for the labels *Approved* and *Rejected*. The cue text is used to describe activities to be actioned by a user in his Tivoli Identity Manager self-care or administrative interfaces. RECERTAPPROVAL has the label *Recertification Approval*.

The Use Notification Text As Action Text check box should be checked. This ensures that the notification text is used as is to describe the approval to be performed in the Tivoli Identity Manager interfaces.

- d. Finally, the last tab of properties. Postscript, allows a JavaScript script to be run immediately after the activity itself has been completed, as shown in Figure 12-55. Its purpose is to perform auditing and to set the result in the event that the activity times out (for example, the user does not process the approval within the specified 7 days).

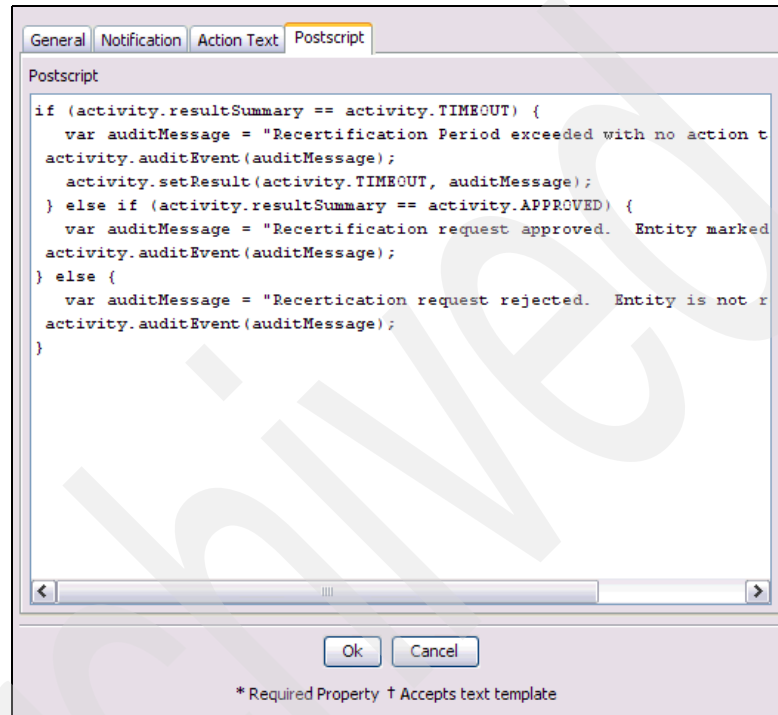


Figure 12-55 Owner approval post-script properties

- e. The manager approval node should be configured identically to the owner approval node, with the following exceptions:
- In the General tab, the activity ID should be different, in this case `MANAGER_RECERT`.
 - In the General tab, the participant should be set to Manager.
 - In the Notification tab, the group email topic property should be set to the same value as the activity ID, in this case `MANAGER_RECERT`.

Figure 12-56 shows the manager approval node's general properties.

ID	Type	Relevant Data ID
entity	Account	Entity
service	Service	theService
owner	Person	Owner

Figure 12-56 Manager approval general properties

At this point in the design of the workflow two approval transition lines should be added.

- a. The first transition line to be defined is the one from the start node to the OWNER_RECERT node, as shown in Figure 12-57 on page 591.

Name	startToOwnerApproval
Description	Start to owner approval node
From	Start (ID: START)
To	\$ITIM_RECERTIFY (ID: OWNER_RECERT)
Condition	<input type="radio"/> Approved <input type="radio"/> Rejected <input checked="" type="radio"/> Custom
<pre>true</pre>	
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>	
<small>* Required Property † Accepts text template</small>	

Figure 12-57 Start to owner approval nodes transition line

Note that the transition line name and description are optional. Also note that the condition field's custom value *true* or an empty value have the same result, which is that the information flows through the transition line in all cases. Conversely, if the condition field contains a script that returns a false value, the information does not flow through the transition line. This is how the information flow can be altered throughout a workflow.

- b. The second transition line to be defined is the one from the OWNER_RECERT to the MANAGER_RECERT node, as shown in Figure 12-58.

Name	OwnerApprovalToManagerApproval
Description	Owner approval node to manager approval node
From	\$ITIM_RECERTIFY (ID: OWNER_RECERT)
To	\$ITIM_RECERTIFY (ID: MANAGER_RECERT)
Condition	<input type="radio"/> Approved <input type="radio"/> Rejected <input checked="" type="radio"/> Custom
<pre>activity.resultSummary == activity.APPROVED</pre>	
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>	
<small>* Required Property † Accepts text template</small>	

Figure 12-58 Owner approval to manager approval nodes transition line

In this transition the following custom condition must be entered:

```
activity.resultSummary == activity.APPROVED
```

This script only allows the flow of information through if its condition returns a *true* value, indicating that the condition has been fulfilled. In this case the script compares the previous activity's result to the specific result code for successful approvals.

The next steps in the configuration covers the workflow's operation nodes. Operation nodes are nodes that call another workflow to be executed. The operation node's activity is finished when the workflow that it has called has completed. The current recertification policy has one such node.

- a. Double-click the ACCOUNT_SUSPEND_OPERATION to handle its properties as shown in Figure 12-59.

The screenshot shows the 'General' tab of a configuration dialog for the 'ACCOUNT_SUSPEND_OPERATION' activity. The fields are as follows:

- * Activity ID: ACCOUNT_SUSPEND_OPERATION
- Activity Name: suspend
- Description: Account suspend operation for recertification
- Join Type: OR
- Split Type: AND
- Operation Activity Type: Data Reference
- Relevant Data: Entity
- Entity Type: Please select entity type
- Entity: Please select entity
- Expression: (empty)
- * Operation: suspend

The 'Input Parameters' section contains a table with the following headers:

ID	Type	Relevant Data ID
----	------	------------------

At the bottom of the dialog are 'Ok' and 'Cancel' buttons, and a note: '* Required Property † Accepts text template'.

Figure 12-59 Account suspend operation general properties

The workflow redirects the process flow to this node if either of the approval activities ends with a rejection. Here the default name (suspend), activity ID (ACCOUNT_SUSPEND_OPERATION), and description are used.

The node join directive is set to *OR*, as the operation only needs one active transition path to reach it to execute.

The next step is to define the operation to call. There are several ways to do this. In this case the Operation Activity Type is set to Data Reference, which identifies entity operations that are defined as *non-static*. The person entity *suspend* operation is one such operation. The relevant data setting identifies the entity for which the operation applies, which in this case is the workflow's parameter called Entity (which represents the account). Then all that is left to do is to select the appropriate operation, *suspend*. Figure 12-59 on page 592 shows the operation node's configuration.

Now that we have added the operation node, two additional transition lines can be configured in the workflow.

- a. The first transition line is the one from the owner recertification approval to the suspend operation. This transition line should only convey the flow of information (be active) when the approval is either rejected or timed out. Therefore, a custom condition should be entered as:

```
activity.resultSummary == activity.REJECTED || activity.resultSummary == activity.TIMEOUT
```

As before, it is not necessary to enter names or descriptions to transition lines, although it may help workflow editors for future customization. Figure 12-60 shows how the transition line is configured in the interface.

Name	ownerApprovalRejectToNonRecertifyOperation
Description	Approval rejection or timeout to the non recertify operation.
From	\$ITIMRECERTIFY (ID: OWNER_RECERT)
To	suspend (ID: ACCOUNT_SUSPEND_OPERATION)
Condition	<input type="radio"/> Approved <input type="radio"/> Rejected <input checked="" type="radio"/> Custom
<pre>activity.resultSummary == activity.REJECTED activity.resultSummary == activity.TIMEOUT</pre>	
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>	
<small>* Required Property † Accepts text template</small>	

Figure 12-60 Owner approval to suspend operation nodes transition line

- b. Figure 12-61 shows how the second transition line, which runs from the manager approval to the operation node, should be configured. Since it is expected to behave in a manner identical to the previous node, the condition governing its behavior will be identically configured:

```
activity.resultSummary == activity.REJECTED || activity.resultSummary == activity.TIMEOUT
```

Name	managerApprovalRejectToNonRecertifyOperation
Description	Approval rejection to the non recertify operation.
From	\$ITIM_RECERTIFY (ID: MANAGER_RECERT)
To	suspend (ID: ACCOUNT_SUSPEND_OPERATION)
Condition	<input type="radio"/> Approved <input type="radio"/> Rejected <input checked="" type="radio"/> Custom
<pre>activity.resultSummary == activity.REJECTED activity.resultSummary == activity.TIMEOUT</pre>	
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>	
* Required Property † Accepts text template	

Figure 12-61 Manager approval to suspend operation nodes transition line

Extension nodes perform specific actions from within the workflow. They are used, amongst other things, to commit provisioning actions set in the workflow. The recertification workflow requires two extension nodes.

- a. The first extension node is used to set an *account suspended* recertification status. The activity ID is set to RECERT_SUSPEND. The activity name is set to SUSPEND. A suitable description can be entered as well.

The join type should be set to *OR*. The leave type should be set to *AND*.

The extension name property represents the name of the extension to be run, complete with the input parameters that it requires. The extension that we need is “recertificationSuspend (Account account)”. Once that extension is selected, the corresponding input parameters appear. In this case, an account-type input must be provided. Clicking **Search Relevant Data** and selecting the account object there completes the definition of the extension node, as shown in Figure 12-62 on page 595.

General Postscript

* Activity ID: RECERT_SUSPEND

Activity Name: SUSPEND

Description: Suspend account extension for recertification

Join Type: AND OR Split Type: AND OR

* Extension Name: recertificationSuspend(Account account)

Input Parameters Search Relevant Data

ID	Type	Relevant Data ID
account	Account	Entity

Output Parameters Search Relevant Data

ID	Type	Relevant Data ID
----	------	------------------

Ok Cancel

* Required Property † Accepts text template

Figure 12-62 Account suspension of recertification extension general properties

- b. The second extension node is used to set an *account recertified* recertification status. Its activity ID and name are both RECERTIFY. A description can be entered.

The join type should be set to *OR*. The leave type should be set to *AND*.

The extension that we need for this node is “recertificationCertify (Account account)”. Once that extension is selected, click **Search Relevant Data** and select the account object there. Figure 12-63 shows the configured account recertification node.

The screenshot shows the 'General' tab of a configuration window. The fields are as follows:

- * Activity ID: RECERTIFY
- Activity Name: RECERTIFY
- Description: Recertify extension for recertification
- Join Type: AND OR
- Split Type: AND OR
- * Extension Name: recertificationCertify(Account account)

Input Parameters table:

ID	Type	Relevant Data ID
account	Account	Entity

Output Parameters table:

ID	Type	Relevant Data ID
----	------	------------------

Buttons: Ok, Cancel

* Required Property † Accepts text template

Figure 12-63 Account recertification extension general properties

To finish configuring the workflow, the remaining transition lines must be entered. Four extension transition lines are required at this point.

- a. The first transition line, as shown in Figure 12-64 on page 597, links the manager approval node to the recertification extension node. This node should only let the information flow through (be active) when the manager approval node activity is completed with a status of *approved*. This can be done in two ways:
 - By entering the following custom condition:


```
activity.resultSummary == activity.APPROVED
```
 - By setting the Condition to *Approved*

Name	approvalApproveToRecertify
Description	Approved the approval
From	\$ITIM_RECERTIFY (ID: MANAGER_RECERT)
To	RECERTIFY (ID: RECERTIFY)
Condition	<input checked="" type="radio"/> Approved <input type="radio"/> Rejected <input type="radio"/> Custom
<pre>activity.resultSummary==activity.APPROVED</pre>	
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>	
<small>* Required Property † Accepts text template</small>	

Figure 12-64 Manager approval to recertification nodes transition line

- b. Finally, three transition lines remain to be entered. These should all have a custom condition set to *true* or an empty value, as there is no condition that must be set for them. These transition lines are between the account suspension operation node to the recertification suspension extension node (as shown in Figure 12-65), the recertification suspension extension node to the end node, and the recertification extension node to the end node.

Name	rejectOperationToNonRecertifyExtension
Description	Rejected the approval
From	suspend (ID: ACCOUNT_SUSPEND_OPERATION)
To	SUSPEND (ID: RECERT_SUSPEND)
Condition	<input type="radio"/> Approved <input type="radio"/> Rejected <input checked="" type="radio"/> Custom
<pre>true</pre>	
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>	
<small>* Required Property † Accepts text template</small>	

Figure 12-65 Account suspension operation

With this configuration, TAA certifies the need for Linux accounts each month. This process will start at 8 a.m. on the second of every month.

6. This concludes the workflow configuration and you can now click **Finish**.

12.5.4 TAA's role and group recertification implementation

In addition to TAA's monthly critical account recertification, TAA will implement an annual role and group recertification policy. This recertification policy executes annually in January of each year.

Managers receive a recertification notification for each employee reporting to them. Using the administrative or self-service interface, managers are able to access the recertification activity for each employee reporting to them. The activity displays, in one view, each employee's groups and roles requiring recertification. The managers should respond *yes* or *no* to whether each role or group membership is still required. Managers can use the *Preview the impact of your selections* feature to review the impact of their recertification responses before submitting the recertification.

Configuration of the recertification policy

To enable the recertification of roles and groups, a recertification policy using a target type of *users* should be defined.

Perform the following steps:

1. Using the administration interface, select **Manage Policies** → **Manage Recertification Policies**. Click **Create** and complete the policies General window, as shown in Figure 12-66.

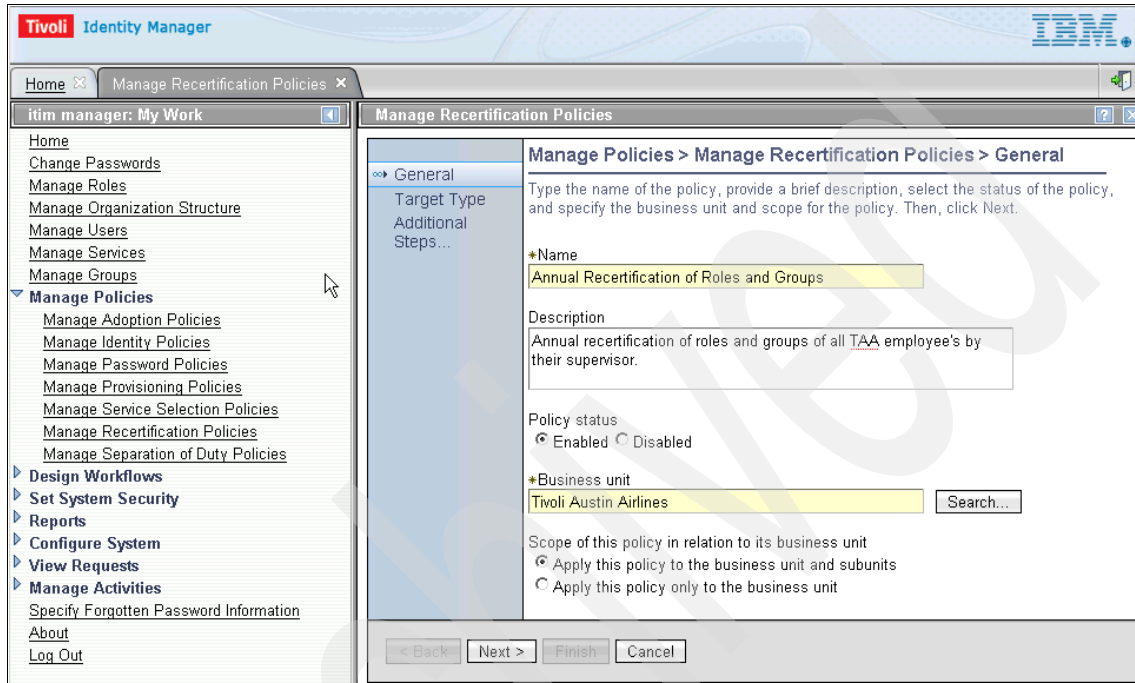


Figure 12-66 Recertification policy: General tab

2. Once the General window has been completed, click **Next** and complete the Target Type window. Select the policy recertification for **Users**, as shown in Figure 12-67.

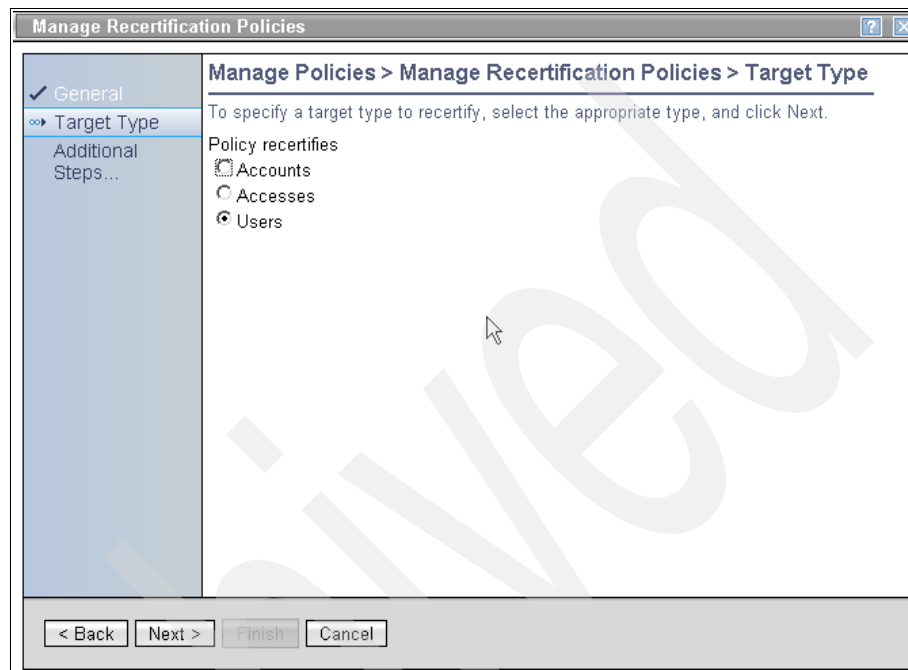


Figure 12-67 Recertification policy: select Target Type

3. Click **Next** to progress to the User Target window and, using the drop-down menu select **person**. Click **Next** to move to the Resource Targets window. Complete the Resource Targets window by selecting All for both the roles and groups, but select **None** for accounts because TAA is only interested in recertifying role and group memberships, but not accounts (Figure 12-68 on page 601).

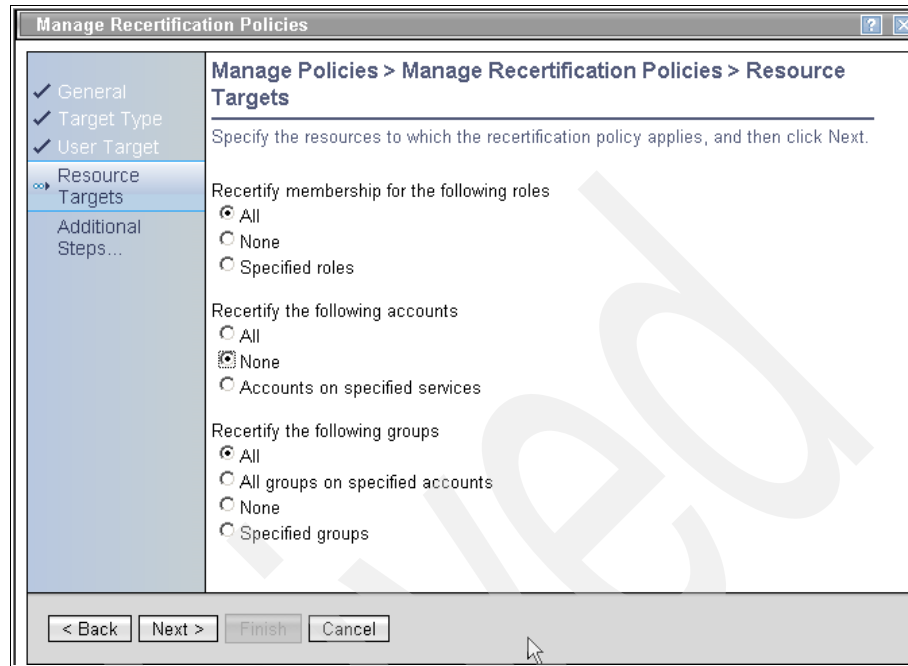


Figure 12-68 Recertification policy: select Resource Targets

4. Once you have completed the Resource Targets window, click **Next** to go to the Schedule window. Create a schedule of type *calendar* to run annually on 10. January at 8:00 AM. Then click **Next** to go to the Policy window.

- The Policy step can be used in two configuration modes, simple and advanced. Previously, during the configuration of the Linux monthly recertification policy that we documented in 12.5.3, “TAA’s critical account recertification implementation” on page 579, the advanced option was used. This was done because customization to add an additional approval step was required. The annual role and group recertification policy requires no additional customization, so the simple configuration wizard can be used this time. Complete the step as shown in Figure 12-69.

The screenshot shows a window titled "Manage Recertification Policies" with a sidebar on the left containing a tree view with the following items: General, Target Type, User Target, Resource Targets, Schedule, Policy (selected), Recertification, E-mail, and Rejection E-mail. The main area is titled "Manage Policies > Manage Recertification Policies > Policy" and contains the following text: "To configure a policy, select Simple and complete the configuration fields, or select Advanced to use the workflow designer. When you are done, click the appropriate button." Below this is the "Configuration mode" section with radio buttons for "Simple" (selected) and "Advanced". The "Who approves recertification" field is a dropdown menu with "Manager" selected. The "Action when recertification is rejected" field is a dropdown menu with "Remove" selected. The "Send rejection e-mail to" field is a dropdown menu with "User being recertified" selected. The "*Days until recertification is due" field is a text input with "14" entered. The "Action when recertification is overdue" field is a dropdown menu with "Approve all" selected. At the bottom of the window are four buttons: "< Back", "Next >", "Finish", and "Cancel".

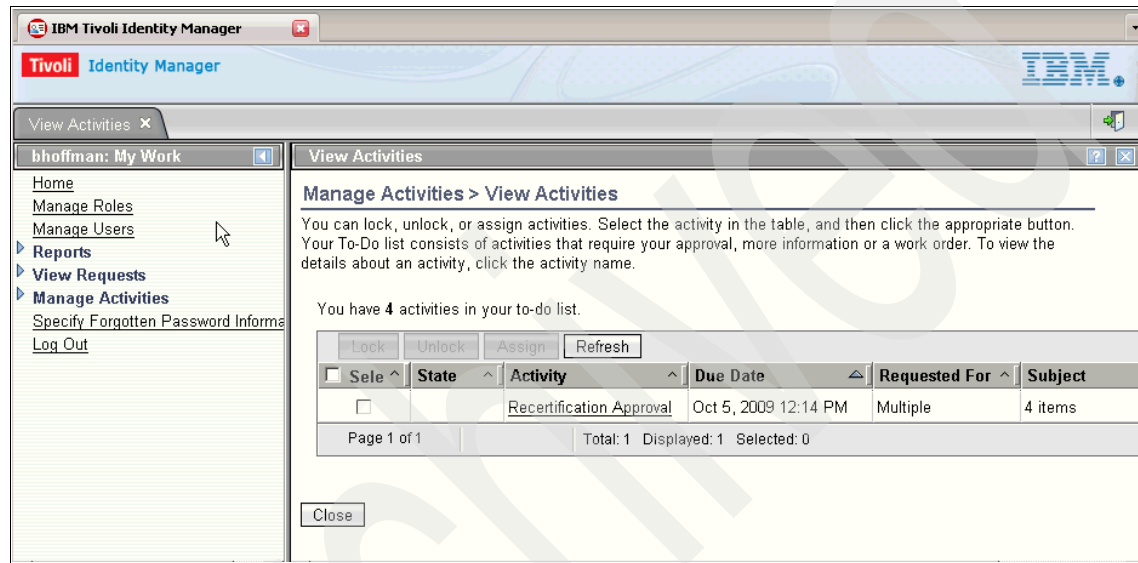
Figure 12-69 Recertification policy: simple policy

- The remaining steps for Recertification E-mail and Rejection E-mail allow for the creation of new customized e-mail templates. For TAA, the default templates are acceptable, so click **Finish** to complete the recertification policy creation.

The recertification policy is now completed and will execute on 10 January each year. Managers have 14 days to respond; after that time the recertifications are automatically approved, which is the TAA chosen action for overdue recertifications of roles and groups.

12.5.5 Responding to recertification activities

In this section, we give an insight into the managers' views of the recertification process. Upon receipt of pending recertification notifications, managers can access either the self-service or administration interfaces. Any pending activity work items are displayed. For examples of pending activities for the TAA manager Robert Hoffman, see Figure 12-70 for the Tivoli Identity Manager administrative interface and Figure 12-71 on page 604 for the self-service interface.



The screenshot shows the IBM Tivoli Identity Manager administrative interface. The browser window title is "IBM Tivoli Identity Manager". The page header includes the Tivoli logo and "Identity Manager". The left sidebar shows a navigation menu with options: Home, Manage Roles, Manage Users, Reports, View Requests, Manage Activities, Specify Forgotten Password Informa, and Log Out. The main content area is titled "View Activities" and "Manage Activities > View Activities". It contains a message: "You can lock, unlock, or assign activities. Select the activity in the table, and then click the appropriate button. Your To-Do list consists of activities that require your approval, more information or a work order. To view the details about an activity, click the activity name." Below this, it states "You have 4 activities in your to-do list." There are buttons for "Lock", "Unlock", "Assign", and "Refresh". A table displays the activity details:

<input type="checkbox"/> Sele ^	State ^	Activity ^	Due Date ^	Requested For ^	Subject
<input type="checkbox"/>		Recertification Approval	Oct 5, 2009 12:14 PM	Multiple	4 items

Page 1 of 1 Total: 1 Displayed: 1 Selected: 0

Close

Figure 12-70 Administration interface: activities pending

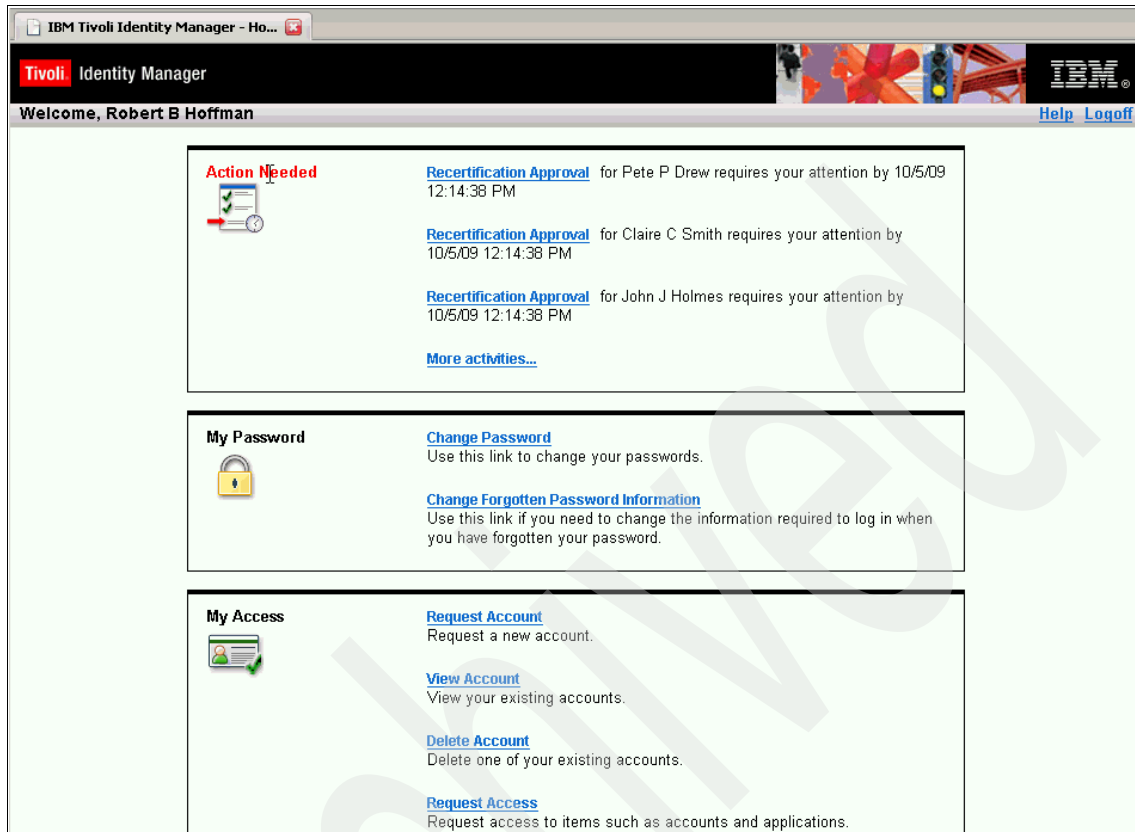


Figure 12-71 Self-service interface: activities pending

Click the activity to expand the details. Robert Hoffman should review the details before submitting a response. If more instructions are required, he can expand the **Instruction Details** section.

After reviewing the details, Robert Hoffman completes the Reviewer Action section by providing either yes or no responses for each role or group being reviewed. Before submitting the responses, he may use the **Preview the impact of your selections** option to review the impact of the changes before submitting them.

Once Robert Hoffman has specified his recertification decisions, he may optionally add reviewer comments and then submit his response. Occasionally he will not be able to complete the recertification in one transaction. For example, he may need to seek additional information before he can complete the recertification. In this situation, Robert can simply save the activity as a draft. Once he is ready to complete the recertification activity, he can access it again, make updates, and then submit it. In Figure 12-72, we show Robert Hoffman's view of the recertification activity for employee John Holmes from the self-service interface.

Welcome, Robert B Hoffman [Help](#) [Logout](#)

[Home](#) > [Approve and review requests](#) > Review request

Review Request

Review the details of this request. To complete this activity, select the appropriate action, enter information in the comments field, and click Submit. To save your selections without completing the request at this time, click Save as Draft. To review other activities without completing the request or saving your selections, click Cancel.

Request Detail

Date submitted: September 21, 2009 12:14:36 PM
 Request type: Recertification Policy
 Requested for: John J Holmes
 Requested by: IBM Tivoli Identity Manager System
 Due date: October 5, 2009 12:14:38 PM
 Instruction summary: Recertification Approval

▶ Instruction Detail

Reviewer Action

Indicate whether or not John J Holmes still requires each of the following roles:

* Please note that all items require a decision

Roles	Description	Still Required	All None
Employee Role	Employee role, this is also a parent role.	<input checked="" type="radio"/> Yes <input type="radio"/> No	
Flight Operations Role	Flight Operations Role	<input checked="" type="radio"/> Yes <input type="radio"/> No	
Mobile Phone User	Mobile Phone User	<input checked="" type="radio"/> Yes <input type="radio"/> No	

Indicate whether or not John J Holmes still requires each of the following accounts and groups:

* Please note that all items require a decision

Accounts and Groups	Description	Still Required	All None
<input type="checkbox"/> jholmes on ITIM Service • Help Desk Assistant	Default Help Desk Assistant Group for Open Financial Network	<input checked="" type="radio"/> Yes <input type="radio"/> No	
<input type="checkbox"/> jholmes on TAA LDAP • FlightOps	TAA LDAP Service	<input checked="" type="radio"/> Yes <input type="radio"/> No	

[Preview the impact of your selections.](#)

Reviewer Comments

Figure 12-72 Example recertification activity

12.6 Future deployment phases

In this section we discuss additional functions, some of which relate to examples shipped by IBM for demonstration purposes with the Tivoli Identity Manager software. During the installation procedures this code is placed in the <ITIM install directory>/extensions/5.1/examples.

12.6.1 Advanced reporting

There are several ways to create reports aside from using the Tivoli Identity Manager custom report functionality provided with the Tivoli Identity Manager administrative console. They include:

- ▶ Hooked reports

Hooked reports are hard-coded reports that can be run from the Tivoli Identity Manager console. These reports are created using XML, JSP, and JAR files. Once installed, they can be viewed in the Tivoli Identity Manager administrative interface like standard reports there.

For more information refer to the <ITIM install directory>/extensions/5.1/examples/adhocreporting directory, where several example files can be found.

- ▶ Integration with Crystal Reports

Tivoli Identity Manager can integrate with Crystal Reports to provide Crystal Reports-driven report templates from its administrative console.

For more information about Tivoli Identity Manager and Crystal Reports integration, refer to the “Configuring” → “Crystal Reports configuration” section in the IBM Tivoli Identity Manager Information Center Version 5.1, at:

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

- ▶ Integration with BIRT

The Business Intelligence and Reporting Tools (BIRT) is an Eclipse-based reporting system integrated with Java/J2EE to produce reports.

A report package can be set up with Tivoli Common Reporting and Tivoli Identity Manager to produce an additional set of sample reports. For more information about Tivoli Common Reporting and BIRT, refer to the “Configuring and administering IBM Tivoli Common Reporting section” in the IBM Tivoli Identity Manager Information Center Version 5.1, found at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/cpt/cpt_ic_reportspack_intro.html

12.6.2 Advanced workflow customization

Tivoli Identity Manager offers the possibility to customize workflows in a number of ways to extend their functionality.

Workflow editing

All Tivoli Identity Manager workflows can be altered to perform additional activities to match business processes. An example, which can be found in the <ITIM install directory>\extensions\examples\workflow\personStatus, is how to customize workflows to allow an HR feed to suspend and restore a person.

Another example, which can be found in the <ITIM install directory>\extensions\examples\workflow\customApproval.html file, describes how a password change operation can be edited to send a notification when the password change operation failed.

Custom processes

In Tivoli Identity Manager workflows, extension nodes allow processes to be executed as part of the workflow to perform operations such as creating accounts or modifying person objects.

It is possible to implement custom processes that can be invoked through extension nodes in the Tivoli Identity Manager workflows.

An example of custom extension process implementation can be found in the <ITIM install directory>\extensions\examples\workflow\Readme.html file.

JavaScript extensions

Within Tivoli Identity Manager, a JavaScript engine allows administrators to define scrips that can be used from defining workflow functionality to setting provisioning policy entitlement parameters.

It is possible to define custom JavaScript extensions that provide the ability to call functions that can perform functionality not provided by the JavaScript engine by default.

Information and examples of how to implement JavaScript extensions, as well as how to invoke them from an Identity manager workflow, can be found in the <ITIM install directory>\extensions\examples\workflow\customApproval.html file.

12.6.3 Adapter customization

Tivoli Directory Integrator adapters, also called RMI-based adapters or custom adapters, are defined by configuration files that are run in Tivoli Directory Integrator via the Remote Method Invocation (RMI) protocol. These configuration files are called AssemblyLines.

It is possible to edit the configuration of existing adapters, as well as create complete adapters. This is done by editing or creating adapter profiles and AssemblyLines. The Tivoli Identity Manager adapter development tool can assist you with adapter development.

A sample adapter, with customization information, can be found in the <ITIM install directory>\extensions\examples\SampleLdapAdapter directory.

Tivoli Identity Manager Adapter Development Tool

This package can be found in the IBM Tivoli Open Process Automation Library (which contains integration examples for Tivoli Identity Manager and other Tivoli solutions). It can be found at the following address:

<http://www.ibm.com/software/brandcatalog/portal/opal/details?catalog.label=ITW1OIMOH>

Figure 12-73 shows a screen capture of the Adapter Development Tool.

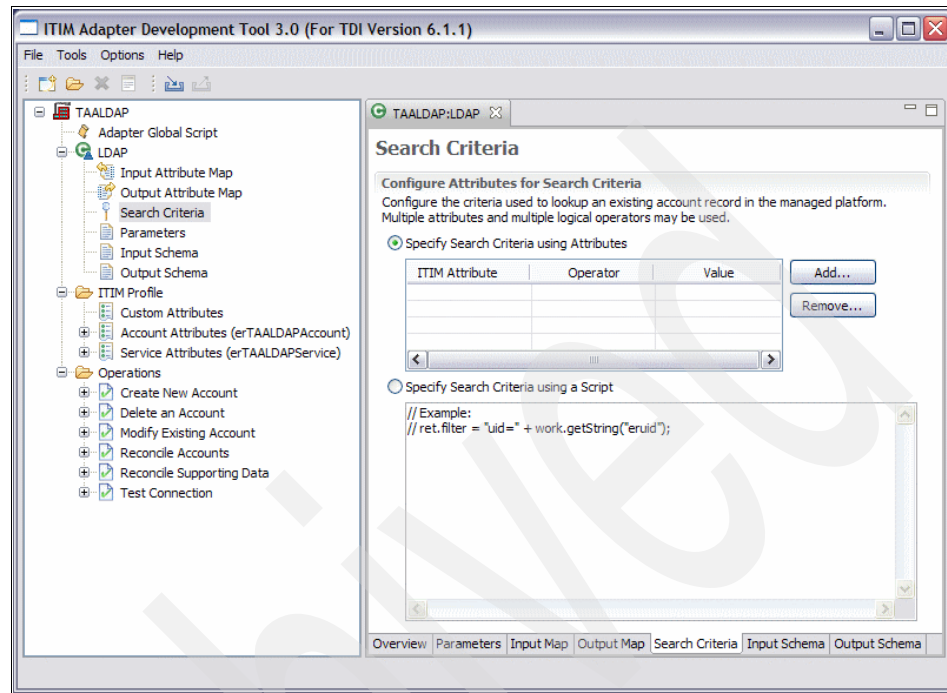


Figure 12-73 Tivoli Identity Manager Adapter Development Tool

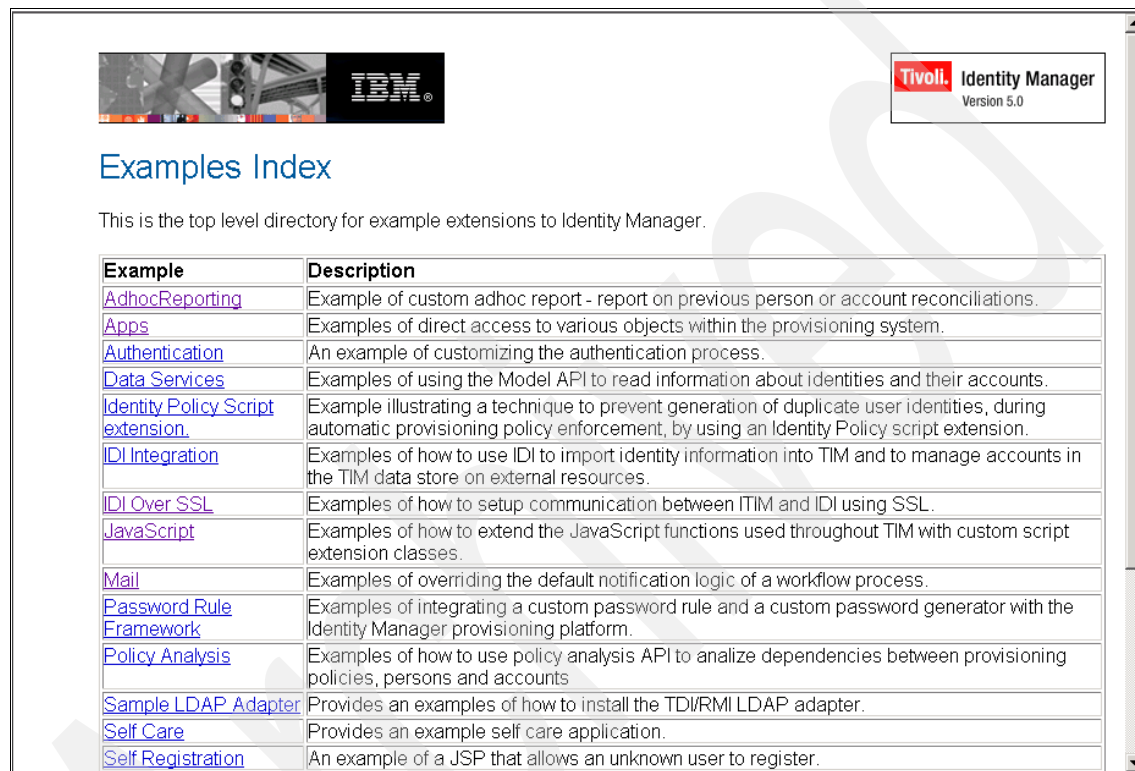
12.6.4 User interface graphics update

The Tivoli Identity Manager administrative console and self-care interfaces' graphical appearances can be customized to give them the *look and feel* of an enterprise's intranet. The administrative console's top and bottom banners can be customized, and the self-care interface's appearance in particular can be customized with cascading style sheets and graphics to integrate with a corporate portal.

For more information about how to customize the Tivoli Identity Manager interfaces' appearances, refer to the "Configuring" → "User interface customization overview" in the IBM Tivoli Identity Manager Information Center Version 5.1.

12.6.5 Additional examples

There are additional examples in <ITIM install directory>/extensions/5.1/examples that may be of interest. Refer to the <ITIM install directory>\extensions\examples\index.html file for more information about Tivoli Identity Manager customization. Figure 12-74 shows an extract of this file.



Example	Description
AdhocReporting	Example of custom adhoc report - report on previous person or account reconciliations.
Apps	Examples of direct access to various objects within the provisioning system.
Authentication	An example of customizing the authentication process.
Data Services	Examples of using the Model API to read information about identities and their accounts.
Identity Policy Script extension.	Example illustrating a technique to prevent generation of duplicate user identities, during automatic provisioning policy enforcement, by using an Identity Policy script extension.
IDI Integration	Examples of how to use IDI to import identity information into TIM and to manage accounts in the TIM data store on external resources.
IDI Over SSL	Examples of how to setup communication between ITIM and IDI using SSL.
JavaScript	Examples of how to extend the JavaScript functions used throughout TIM with custom script extension classes.
Mail	Examples of overriding the default notification logic of a workflow process.
Password Rule Framework	Examples of integrating a custom password rule and a custom password generator with the Identity Manager provisioning platform.
Policy Analysis	Examples of how to use policy analysis API to analyze dependencies between provisioning policies, persons and accounts
Sample LDAP Adapter	Provides an examples of how to install the TDI/RMI LDAP adapter.
Self Care	Provides an example self care application.
Self Registration	An example of a JSP that allows an unknown user to register.

Figure 12-74 Tivoli Identity Manager examples index

12.7 Conclusion

This concludes our extensive business scenario at Tivoli Austin Airlines. We have seen how to effectively deploy a large Tivoli Identity Manager implementation by partitioning it into four phases.

Phase 1: Installation and setup

We created an operational Identity Manager installation with all of the components operational and an accurate list of TAA employees and their accounts imported into the Tivoli Identity Manager directory and organization chart. The tasks in this phase included:

- ▶ Installation of Identity Manager and its required middleware components
- ▶ Definition of any custom person types
- ▶ Creation of an identity feed and validation of the feed data
- ▶ Installation of Identity Manager adapters
- ▶ Execution of reconciliations of each installed adapter to create a list of accounts and mapping to the owners
- ▶ Cleanup of any orphan accounts produced by the reconciliations (required for SOX compliance)
- ▶ Security hardening of the Identity Manager servers and components

Phase 2: Automatic account management

In this phase we generated a quick return on investment for TAA by implementing tasks that addressed high-priority requirements. One of the factors governing how quickly a feature can be implemented and placed into production use is the number of people who must be trained to use the new feature. The implementation tasks for this phase were chosen because they are largely invisible to most employees. Very little training was necessary beyond those people who are Identity Manager administrators. The features included in this phase are:

- ▶ Creation of common accounts (such as e-mail and Windows) for new employees
These accounts will be created automatically when a new person is created by the identity feed.
- ▶ Suspension of a person's accounts when the person is terminated
These accounts will be suspended automatically when the identity feed changes a person's status to an inactive value.
- ▶ Password synchronization using the Windows password change interceptor
When a user changes her Windows password, it will automatically change her other account passwords, too.

Phase 3: Delegated account management

In this phase we delegated account management activities to persons other than system administrators. This phase required the preparation of documentation and training for the delegated administrators. The implementation of these features also required more extensive requirements gathering than the previous phases. The features included in this phase are:

- ▶ Password self-reset using challenge/response questions
This feature was quick and easy to implement, and it resulted in a very large cost savings. But it required giving all users Identity Manager accounts and teaching the users how to set their challenge answers and how to use the feature to reset their passwords.
- ▶ Account management using the Identity Manager Web user interface
This enabled Identity Manager to maintain a centralized audit trail regardless of whether the account management is being done by system administrators or by delegated administrators.
- ▶ Delegation
 - Account management
Users have been enabled to request the creation, modification, and deletion of accounts owned by persons that they supervise.
 - Policy compliance
A region's security and compliance team is notified when users who transfer from one region to another may no longer be entitled to some of their accounts. They are able to decide how long such a user will be allowed to retain his non-compliant account.
 - Group management
Tivoli Identity Manager administrators or service owners now have the ability to manage groups directly. This includes creating and deleting groups. Previously, this could only be done by the local system administrator of the managed resource. This allows for delegation of group management from the managed resource to Tivoli Identity Manager.
- ▶ Change control for the Identity Manager configuration
The development of the phase 3 features did not impact the correct functioning of the phase 2 deployment because the new feature development was conducted in an isolated environment. New features were migrated into the production environment as working units.

Phase 4: Role-based account management

This phase implemented the automatic granting and removing of access rights based on a person's job role. This greatly reduced the chance for errors when granting and auditing access rights.

The features implemented by TAA in this phase were:

- ▶ Definition of roles and policies for multiple access rights
- ▶ Definition of roles and policies for generic system and application access
- ▶ Definition of accesses for generic system and application access
- ▶ Definition of role relationships and hierarchy structures
- ▶ Definition of separation of duty policies

Implementation of a monthly recertification process to recertify accounts on resources regarded as critical by TAA. Employee role and group membership annual recertification by employee's manager.

Archived



Account management workflow customization

In this appendix, we take a closer look at how to customize the workflow for the *delete account* functionality.

Requirements

When TAA employees leave the company, their accounts are suspended immediately. After another three months these accounts are automatically deleted, as described in 10.5, “Account suspension on termination” on page 443.

However, TAA does not want this policy to be applied for their Linux platform. Employees may have stored important data on this system that TAA does not want to lose. Therefore, administrators need to define an extended *delete account* workflow functionality that requires the Linux service owners to approve account deletions after verifying that all important data has been either saved or transferred to a different account. At this time the deletion of Linux accounts has to be manually initiated by Tivoli Identity Manager administrators.

Design considerations

In this appendix we describe what it needs to design and create a workflow of a *delete account* functionality using Tivoli Identity Manager’s graphical workflow design interface.

From the navigation tree, we select, as shown in Figure A-1, Configure System → Manage Operations. Clicking **delete** on the Manage Operations page opens the operation diagram page to customize the related workflow. Make sure to click the lowercase **delete** in the Operation column and *not* the **Delete** button.

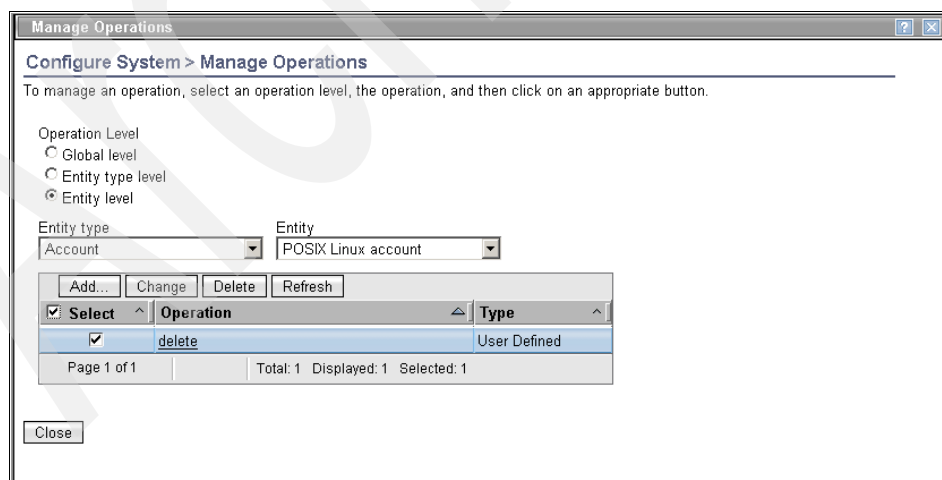


Figure A-1 Manage operations: Delete

Let us step through the operation diagram shown in Figure A-2.

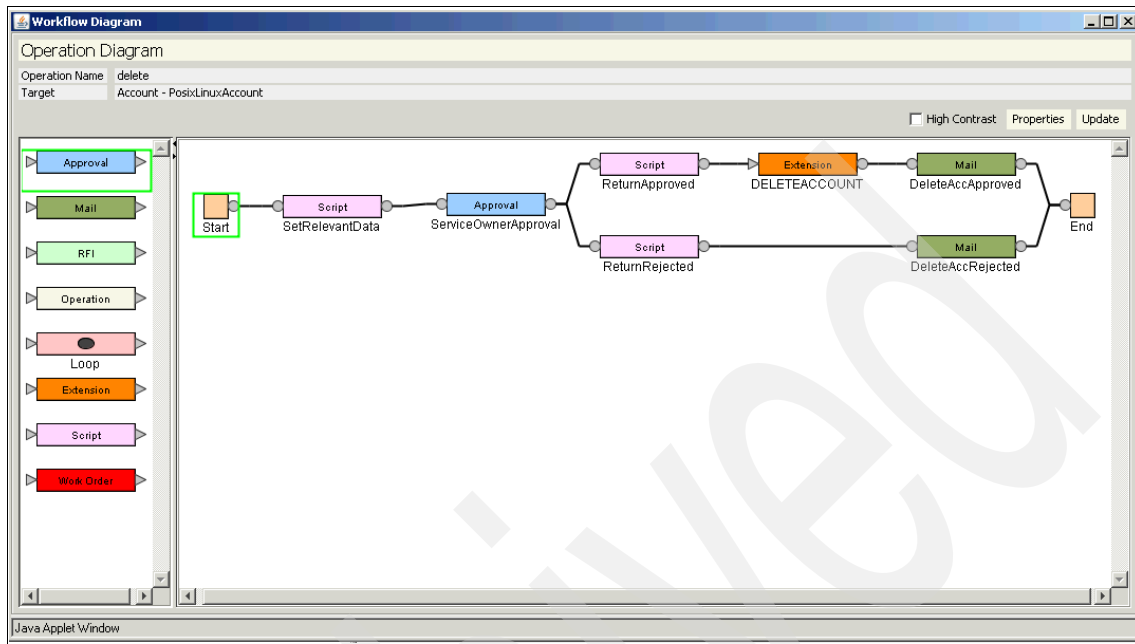


Figure A-2 Delete account workflow

1. The start node is the initiation point of the workflow.
2. In the first script node service and service owner parameters are included for further processing.
3. In the approval node the service owner must approve the request for the entitlement, in our case a *Linux delete account*. The service owner is notified via e-mail that a delete account request has been created for him. One of two decisions can be made, they are depicted in the following two parallel tasks.
 - a. If the service owner approves the delete request the orange workflow extension continues with the deletion of the account, and the following mail node notifies the administrator who requested the delete account about the successful execution.
 - b. If the service owner does not approve the delete request the lower portion of the workflow task continues with the mail node to notify the administrator that the delete account request was rejected.
4. Both tasks complete with the same end node.

Table A-1 identifies the workflow node properties and their values for the workflow named *delete*.

Table A-1 Delete workflow parameters

Node	Feature	Value
Start	Activity ID	Start
	Join Type	AND
	Split Type	AND
Script	Activity ID	SetRelevantData
	Join Type	AND
	Split Type	AND
	Java Script	<pre>var acc = Entity.get(); var servicedn = acc.getProperty("erservice")[0]; var ownerdn = acc.getProperty("owner")[0]; var svc = new Service(servicedn); var per = new Person(ownerdn); service.set(svc); owner.set(per);</pre>
Approval	Activity ID	ServiceOwnerApproval
	Participant	Custom
	Escalation Participant	Participant Type
	Escalation Limit	2 days
	Join Type	AND
	Split Type	AND
	Entity Type	Account
	Approval Code	AA
	Rejected Code	AR

Node	Feature	Value
Script	Activity ID	ReturnApproved
	Join Type	AND
	Split Type	AND
	Java Script	WorkflowRuntimeContext.setProcessResult("AA")
Script	Activity ID	ReturnRejected
	Join Type	AND
	Split Type	AND
	Java Script	WorkflowRuntimeContext.setProcessResult("AR")
Extension	Activity ID	DELETEACCOUNT
	Join Type	AND
	Split Type	AND
	Extension Name	deleteAccount(Accountaccount)
	Postscript	WorkflowRuntimeContext.setProcessResult(WorkflowRuntimeContext.getActivityResult()); WorkflowRuntimeContext.setProcessResultDetail(WorkflowRuntimeContext.getActivityResultDetail());
Mail	Activity ID	DeleteAccRejected
	Recipient	Requestor
	Join Type	AND
	Split Type	AND
	Notification	<RE key="delete_account_request_approved"><PARM><JS>process.subject; </JS></PARM><PARM><JS>var acct=Entity.get();acct.getProperty("service")[0].name;</JS></PARM></RE>

Node	Feature	Value
Mail	Activity ID	DeleteAccApproved
	Recipient	Requestor
	Join Type	AND
	Split Type	AND
	Notification	<RE key="delete_account_request_rejected"><PARM><JS>process.subject;</JS></PARM><PARM><JS>var acct=Entity.get();acct.getProperty("service")[0].name;</JS></PARM></RE>

TAA's implementation

In the following scenario, the Tivoli Identity Manager administrator *ITIM* requests a deletion of TAA employee Robert. B. Hoffman's account, *bhoffman* on service *Linux.Central Region*. As soon as the request is created *Harry Gold*, the owner of the *Linux.Central Region* service is notified of the request. He can then approve or reject the request. Dependent on Harry Gold's decision the administrator *ITIM* is notified whether the request was approved or rejected.

In the following steps, we describe this scenario in more detail.

1. The administrator logs in to the administrator console as user *ITIM*.

2. He selects, in the navigation tree, **Manage Services** → **Select a Service**, as shown in Figure A-3.

Manage Services > Select a Service

To locate a service that you want to manage, type information about the service in the field, select a service type, and then click Search. The services that match your criteria are displayed in the table below. By default, clicking Search will search the system based on the beginning letters of the item you are searching for. To search for a textual pattern in the middle of an item, use the "*" symbol on the keyboard to indicate a wildcard. (For example, typing "b*" will find "abc".)

Search information: * Search by: Service Service type: All Search

Services

To perform a particular task on a service, click the icon next to the service name, and then select the task you want to perform.

18 results found for: *

<input type="button" value="Create"/> <input type="button" value="Change"/> <input type="button" value="Delete"/> <input type="button" value="Refresh"/>				
<input type="checkbox"/> Select	Service Name	Description	Service Typ	Business Un
<input type="checkbox"/>	HiFeed	Daily feed of HR data	IDI data feed	Tvoli Austin Airlines
<input type="checkbox"/>	ITIM Service		ITIM	Tvoli Austin Airlines
<input checked="" type="checkbox"/>	Linux.CentralRegion	Linux Service for users located in the Central Region	POSIX Linux profile	CenterRegion
<input type="checkbox"/>	Linux.CorporateHQ	Linux Service for users located in the Corporate HQ	POSIX Linux profile	AD CorpHQ

Figure A-3 Service: Linux.Central Region

- On the Select a Service page, he selects the **Linux.Central Region** service. By clicking the twisty, the menu shows different possible functions, as shown in Figure A-4.

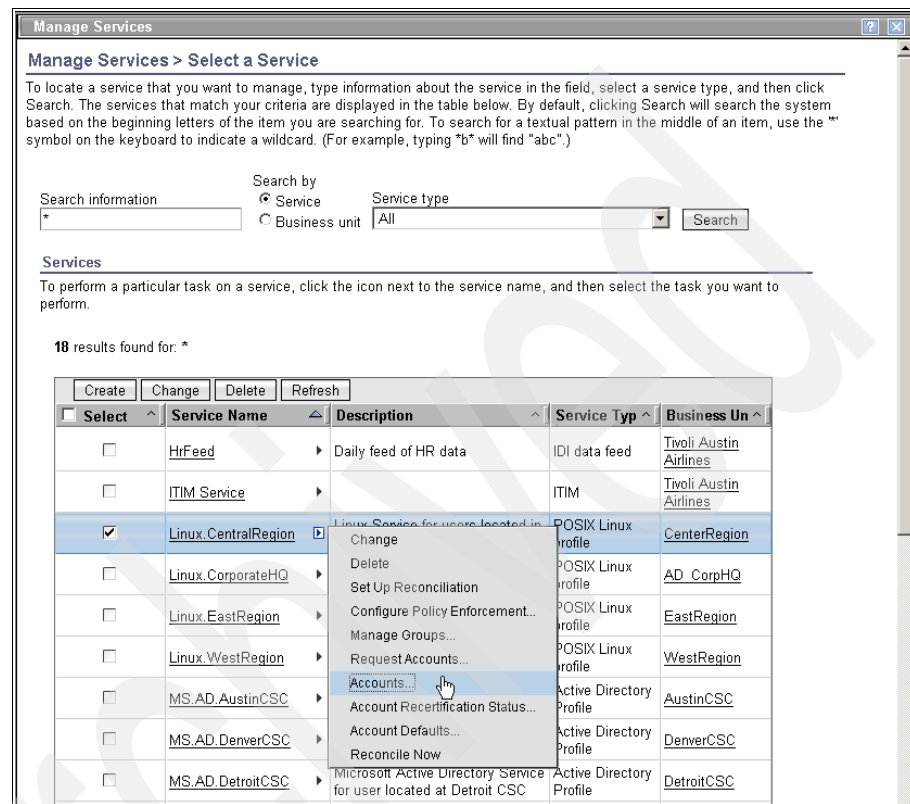


Figure A-4 Select Accounts...

- The administrator clicks **Accounts...** in order to retrieve a list of all related accounts (Figure A-5 on page 623).

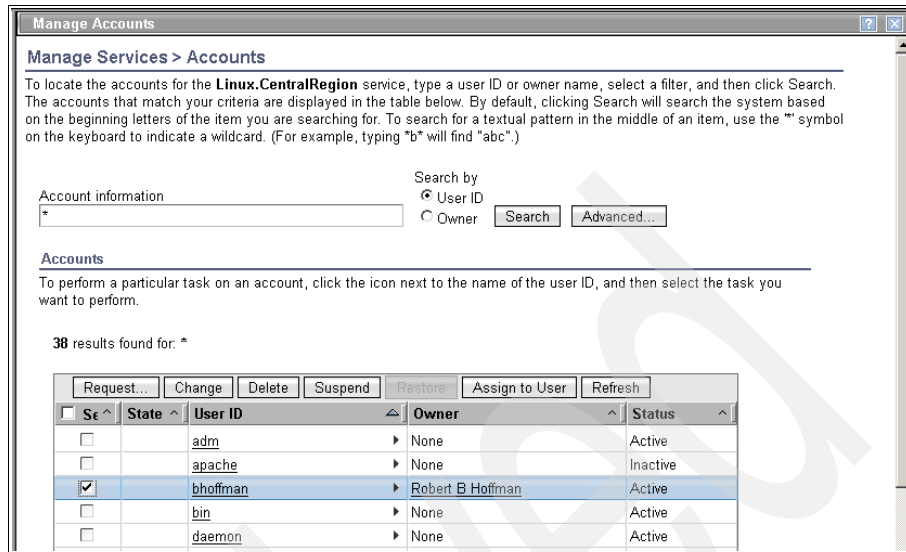


Figure A-5 Account bhoffman

- The administrator selects Robert B. Hoffman's account *bhoffman* and clicks **Delete**.
- On the Confirm page, shown in Figure A-6, he clicks **Delete**.

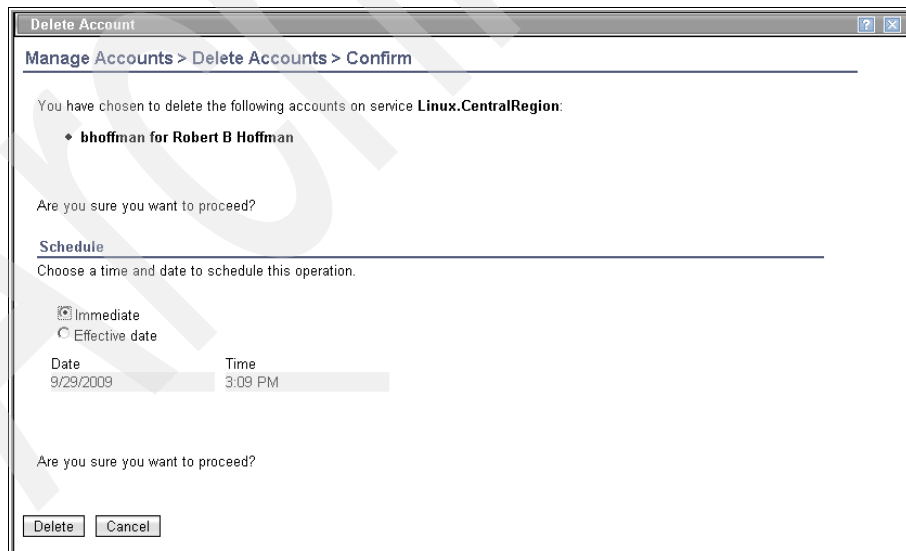


Figure A-6 Delete account bhoffmann

7. On the following success page, shown in Figure A-7, he clicks **Close**.

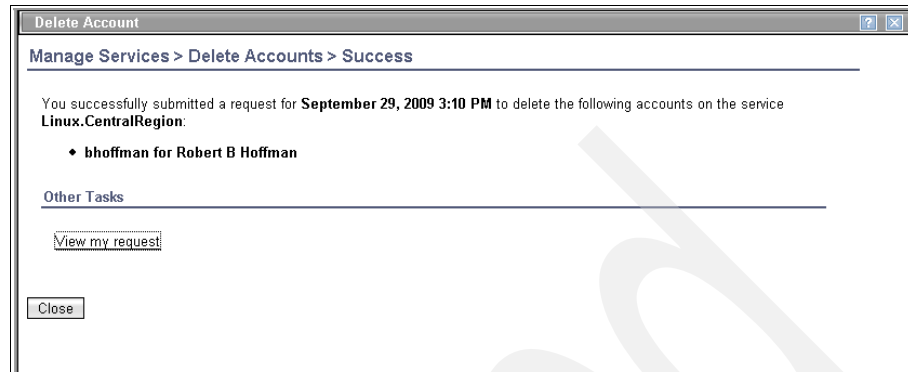


Figure A-7 Success page

8. Now an e-mail is automatically sent to Harry Gold, service owner of the Linux.Central Region service, to seek his approval for the delete account request, as shown in Figure A-8.

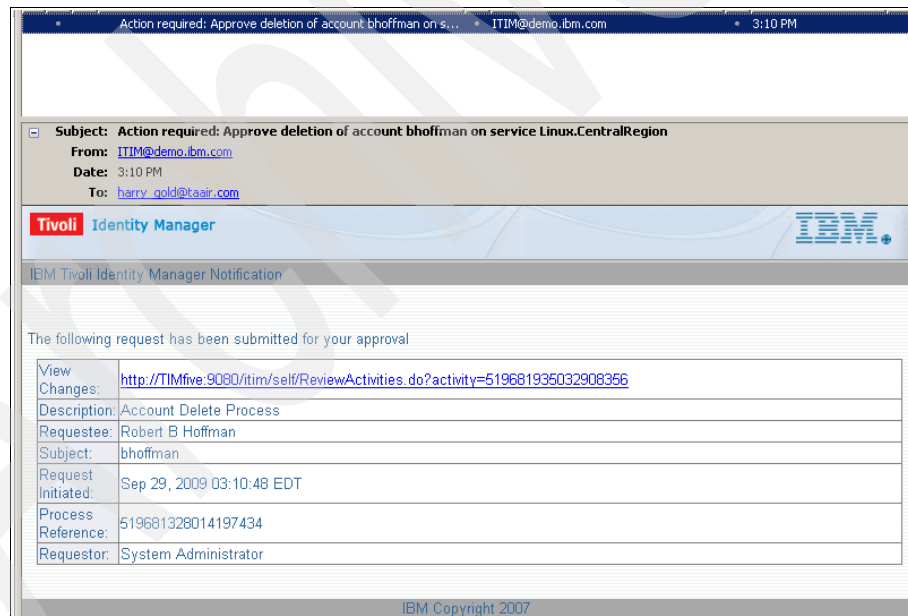


Figure A-8 Harry Gold notification

- The following steps show how Harry Gold approves or rejects the delete account request. From the navigation tree, he selects **Manage Activities** → **View Activities by User** → **Select a User**. Then he selects his user ID *hgold* and continues to the Select a Account page, shown in Figure A-9.

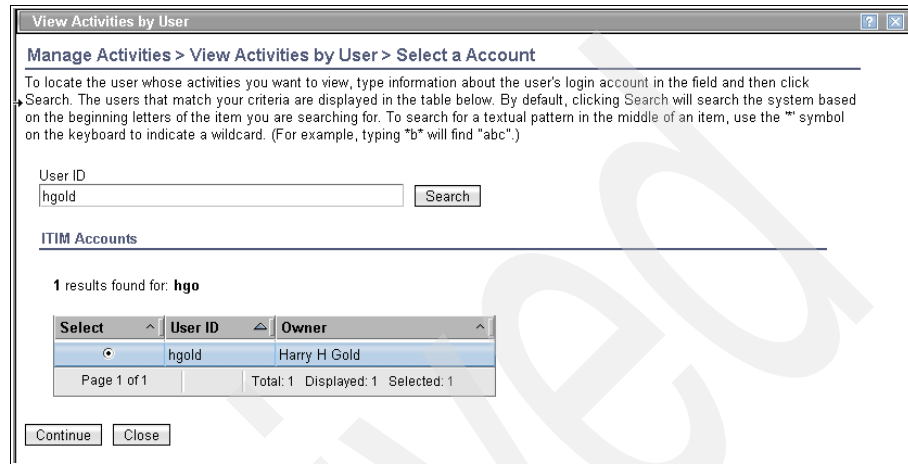


Figure A-9 Select account page

- He clicks **Continue** and gets to the View Activities by User window (Figure A-10).

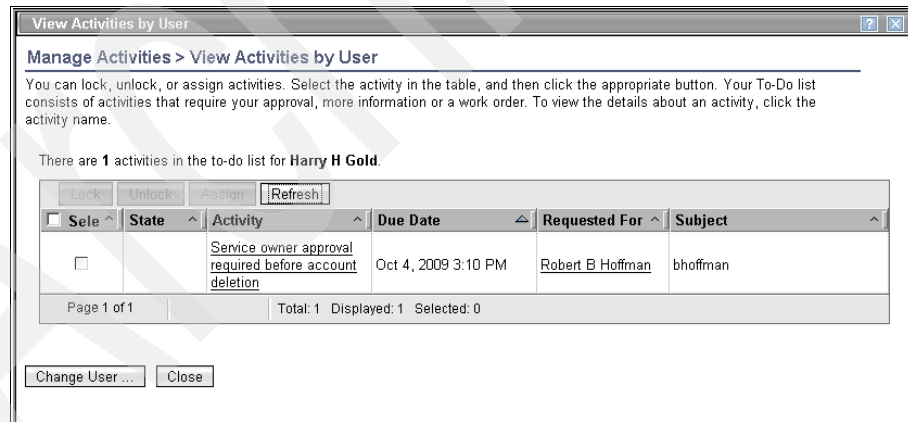
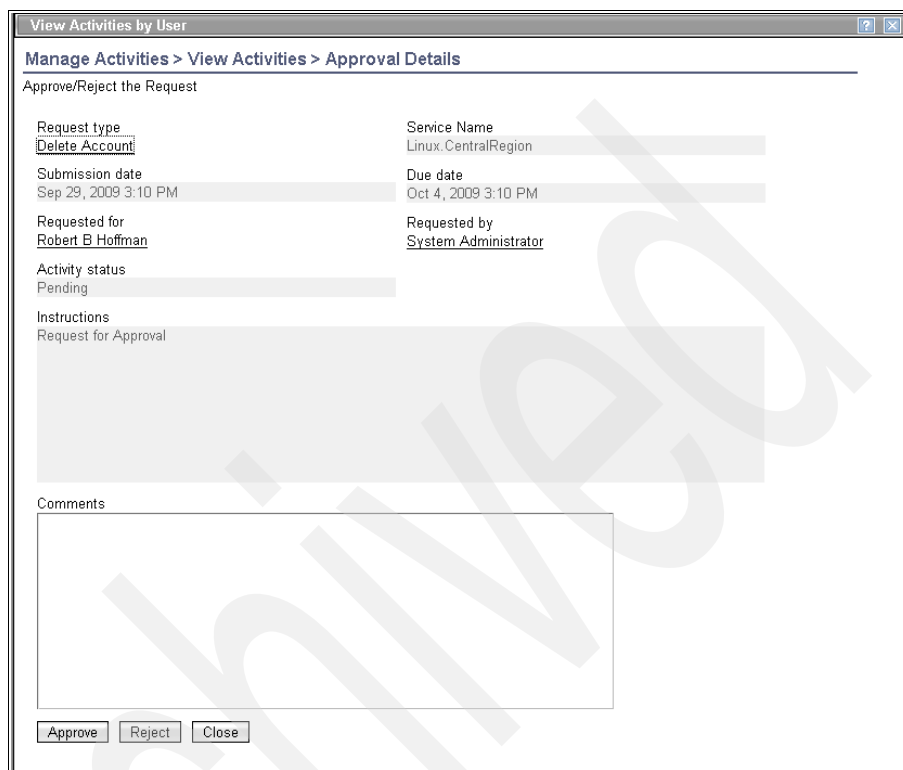


Figure A-10 Activities: User hgold

11. Next, he clicks **Service owner approval required before account deletion**. Figure A-11 shows the approval details.



The screenshot shows a window titled "View Activities by User" with a breadcrumb trail: "Manage Activities > View Activities > Approval Details". Below the breadcrumb is a section titled "Approve/Reject the Request".

Request type Delete Account	Service Name Linux.CentralRegion
Submission date Sep 29, 2009 3:10 PM	Due date Oct 4, 2009 3:10 PM
Requested for Robert B Hoffman	Requested by System Administrator

Activity status
Pending

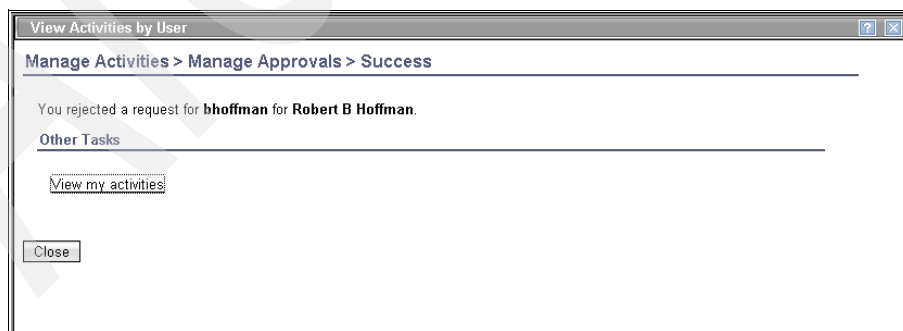
Instructions
Request for Approval

Comments

At the bottom of the window are three buttons: "Approve", "Reject", and "Close".

Figure A-11 Approval details

12. He now clicks **Reject**. Figure A-12 shows the result.



The screenshot shows the same window titled "View Activities by User" with a breadcrumb trail: "Manage Activities > Manage Approvals > Success".

You rejected a request for **bhoffman** for **Robert B Hoffman**.

Other Tasks

[View my activities](#)

[Close](#)

Figure A-12 Reject request

13. An e-mail (Figure A-13) is sent to administrator ITIM to notify him that the request has been rejected.

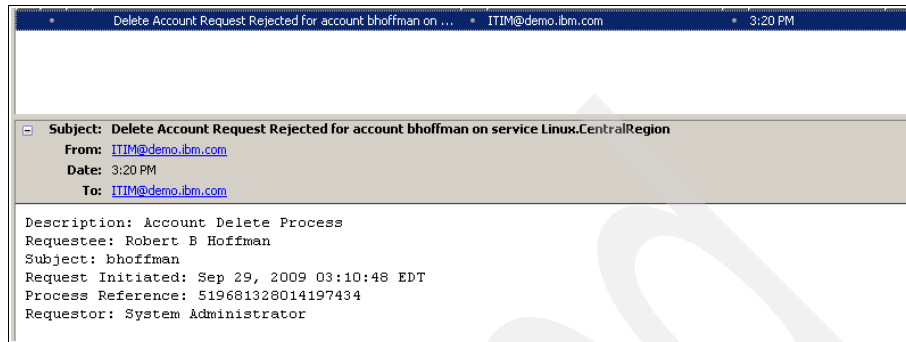


Figure A-13 ITIM notification: request rejected

14. Administrator ITIM again requests a deletion of TAA employee Robert. B. Hoffman's account *bhoffman* on service *Linux.Central Region*. Again, he has to follow steps 1 to 7 as described above.
15. Another e-mail is sent to Harry Gold, service owner of the *Linux.Central Region* service to ask his approval for the delete account request.

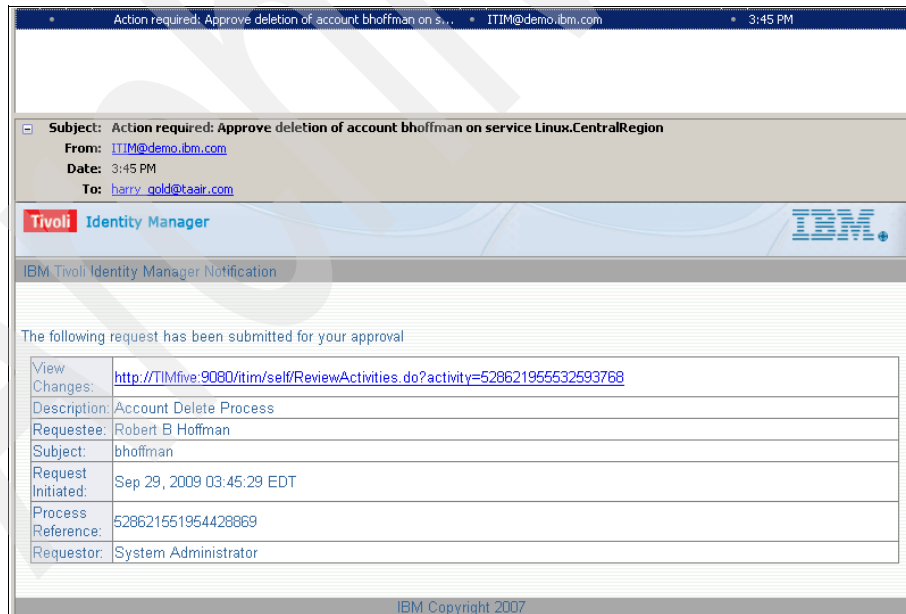
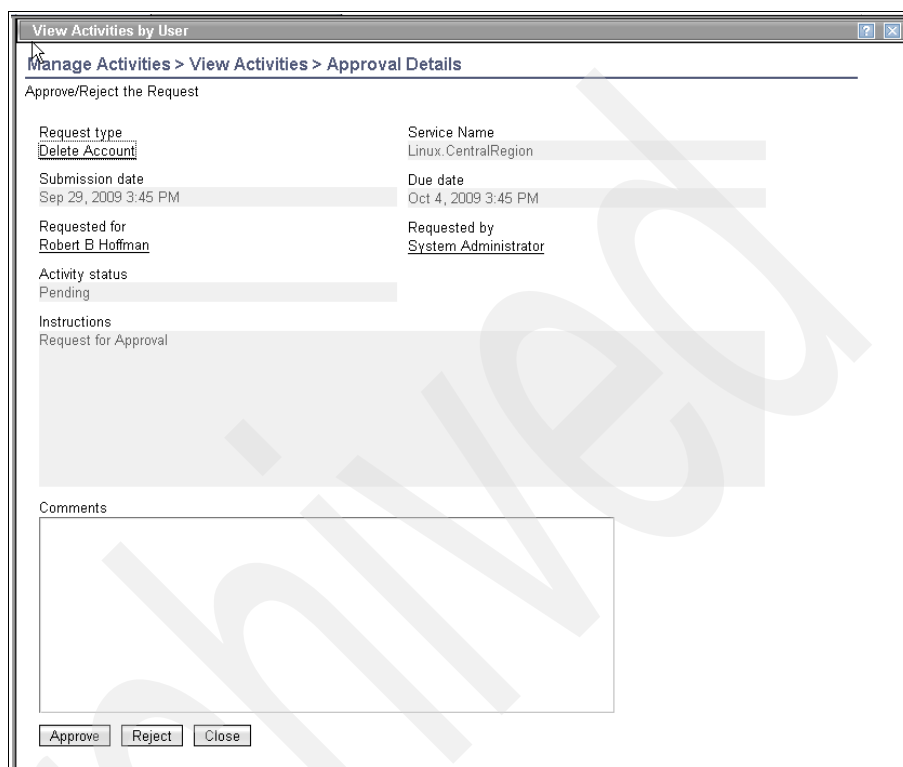


Figure A-14 Request notification for Harry Gold

16. The steps required to approve or reject the account delete request are the same as described in steps 8 to 10 above, and H. Gold gets to the approval details page, as depicted in Figure A-15.



The screenshot shows a web browser window titled "View Activities by User". The breadcrumb navigation is "Manage Activities > View Activities > Approval Details". The main heading is "Approve/Reject the Request".

Request type <u>Delete Account</u>	Service Name Linux:CentralRegion
Submission date Sep 29, 2009 3:45 PM	Due date Oct 4, 2009 3:45 PM
Requested for <u>Robert B Hoffman</u>	Requested by <u>System Administrator</u>

Activity status
Pending

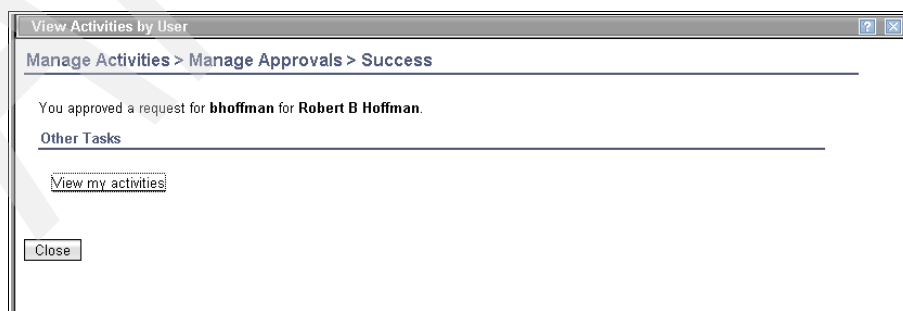
Instructions
Request for Approval

Comments

At the bottom, there are three buttons: "Approve", "Reject", and "Close".

Figure A-15 Approval details

17. He clicks **Approve** and gets the related success page (Figure A-16). On this page, he clicks **Close**.



The screenshot shows the same browser window, but the breadcrumb navigation is "Manage Activities > Manage Approvals > Success".

You approved a request for **bhoffman** for **Robert B Hoffman**.

Other Tasks

[View my activities](#)

Close

Figure A-16 Request approved

18. Another e-mail is sent to administrator ITIM to notify him that the request is approved (Figure A-17).

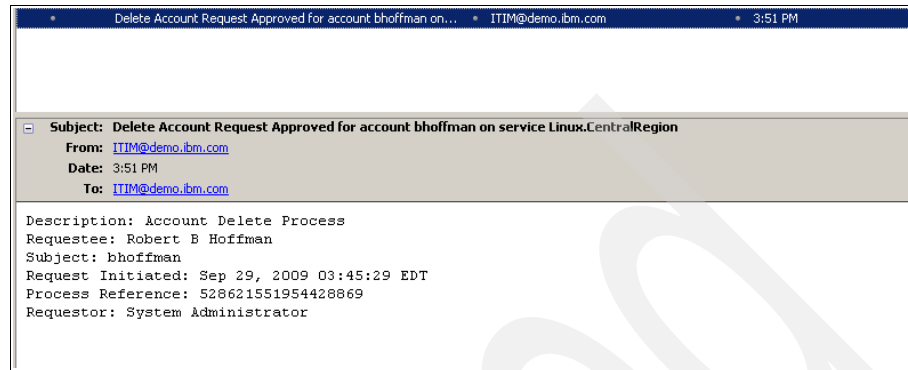


Figure A-17 ITIM notification: request approved

Conclusion

In this appendix we presented an overview of what it needs to customize the workflow for the *delete account* functionality of Linux accounts. We also showed what it needs to design and create this workflow using Tivoli Identity Manager's graphical workflow design interface. Finally, we showed in a scenario how the delete account workflow is used in TAA's environment.

Archived

Windows desktop password reset and unlock

In this appendix we take a closer look at the Windows desktop password reset and unlock functionality. For more information about the IBM Tivoli Identity Desktop Password Reset Assistant, refer to IBM Tivoli Identity Manager Information Center Version 5.1.

Requirements

As discussed in 8.2, “Functional requirements” on page 325, users are supposed to have the ability to reset their own passwords using the Tivoli Identity Manager self-service interface. TAA need to address the functional requirements shown in Table B-1.

Table B-1 Functional requirements for timely password management

Requirement	Description
A	Users will have a single password for all of their accounts.
B	Password resets will be delegated to users other than the system administrators (possibly to the users).

Let us have a quick look at the current password reset and unlock situation.

After finishing implementation phase II, users can reset their passwords using the Tivoli Identity Manager Password Synchronization for Active Directory Plug-in as shown in 10.4, “Password synchronization using the Windows password interceptor” on page 436. When using the Tivoli Identity Manager Password Synchronization for Active Directory Plug-in the users can change their password using the regular Windows user interface, then the adapter sends a password synchronization request to Tivoli Identity Manager for all accounts that the users own.

In implementation phase III all users are provisioned with a Tivoli Identity Manager account. When users log on to the Tivoli Identity Manager self-service interface for the first time, they have to provide additional personal information in order to use the password reset capability. This information will be required in case users forget their passwords and they want to reset it using the Tivoli Identity Manager interface.

There are still some situations that can cause users to call the help desk to reset or unlock their Windows passwords. This situation can occur if users cannot access their Windows desktop, and thus have no access to a Web browser in order to utilize the Tivoli Identity Manager self-service user interface for a password reset. The two major situations that we are addressing are a Windows account being unlocked and a Windows domain password reset. The Tivoli Identity Manager self-service user interface is capable to resolve both situations, but since the users have no access to a Web browser, the Microsoft Graphical Identification and Authentication (GINA) module is the only interface available for the user in this situation.

Design considerations

In order to address the Windows password reset problem, TAA has decided to implement the Tivoli Identity Manager Desktop Password Reset Assistant on all systems running Microsoft Windows XP and Vista. This tool allows users to reset their passwords and unlock their Windows AD accounts.

There are two supported configurations for the Tivoli Identity Manager Desktop Password Reset Assistant. The first configuration is the basic configuration in which Tivoli Identity Manager communicates with the Desktop Password Reset Assistant software that is installed on each user's desktop computer. The second configuration is an enhanced solution that uses the basic configuration with optional Tivoli Access Manager for Enterprise Single Sign-On components.

Since TAA has not yet implemented any Tivoli Access Manager for Enterprise Single Sign-On products, the basic configuration will be implemented. For more details about the basic configuration refer to *Tivoli Identity Manager Version 5.1 Desktop Password Reset Assistant Installation and User Guide*, SC23-9625.

TAA's implementation

TAA needs to install the Tivoli Identity Manager Desktop Password Reset Assistant (DPRA) on every desktop machine running Microsoft Windows XP or Vista. A complete list of supported operating systems is available in *Tivoli Identity Manager Version 5.1 Desktop Password Reset Assistant Installation and User Guide*, SC23-9625.

Note: For large enterprise deployments you should use an automated software provisioning product such as Tivoli Provisioning Manager or Microsoft SMS.

Before beginning the deployment of the software component it is necessary to obtain additional information. Table B-2 shows the information required.

Table B-2 Configuration values for desktop password reset adapter

Field	Explanation	TAA's values
Tivoli Identity Manager server host name or IP address	Specifies the host name or IP address of the Tivoli Identity Manager server	timfive
Tivoli Identity Manager SSL port number	Specifies the SSL port number of Tivoli Identity Manager server	9443
Certificate file	The file containing the digital certificate of the CA for the Tivoli Identity Manager server	c:\timfive.der

Note: Only one full GINA module can exist on a Windows system at any given time. Any vendor product that installs its own GINA replacement module must be installed on the Windows desktop *before* installing the Tivoli Identity Manager Desktop Password Reset Assistant because the Tivoli Identity Manager adapter *adds* functionality to an existing GINA. If the adapter is installed first, however, installing a vendor's GINA also replaces the Desktop Password Reset Assistant. You must click **Yes** when prompted to replace the current GINA (ItimCRGina.dll) with the vendor product GINA, then reinstall the Desktop Password Reset Assistant.

DPRA installation

When installing the DPRA using the graphical interface, follow the next steps:

1. Start the installation program using the SetupGina.exe file in the directory where you have the software. For example, select **Run** from the Start menu, and type C:\TEMP\SetupGina.exe in the Open field.
2. In the Welcome window, click **Next**.
3. In the License Agreement window, review the license agreement and decide if you accept the terms of the license. If you do, select Accept and then click **Next**.

4. Enter the Tivoli Identity Manager server Host name or IP address and Tivoli Identity Manager SSL port number as shown in Figure B-1, then click **Next**.

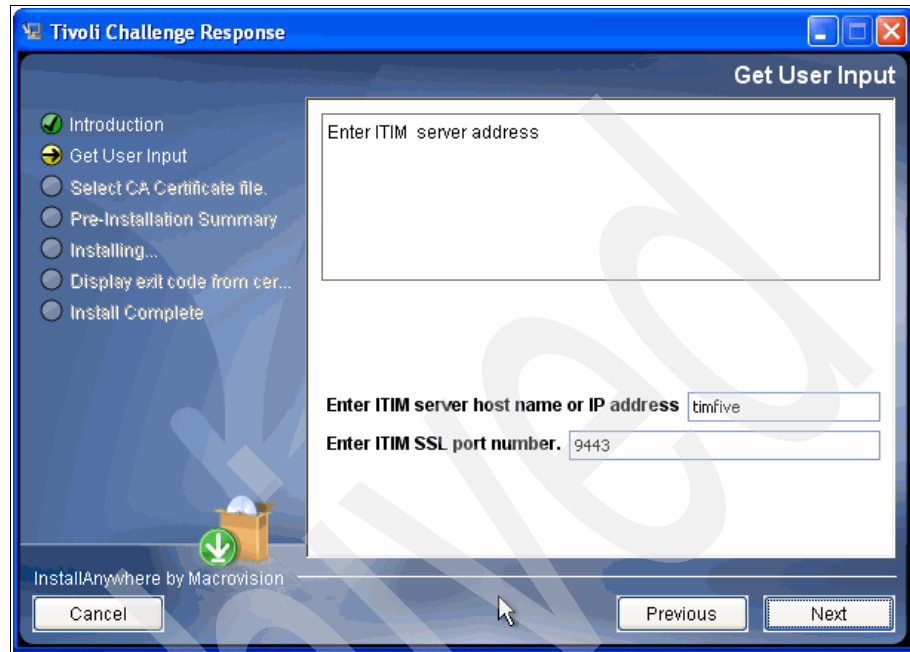


Figure B-1 Tivoli Identity Manager server information

5. Select a certificate file from your Tivoli Identity Manager server by using the **Choose** button, as shown in Figure B-2, then click **Next**.

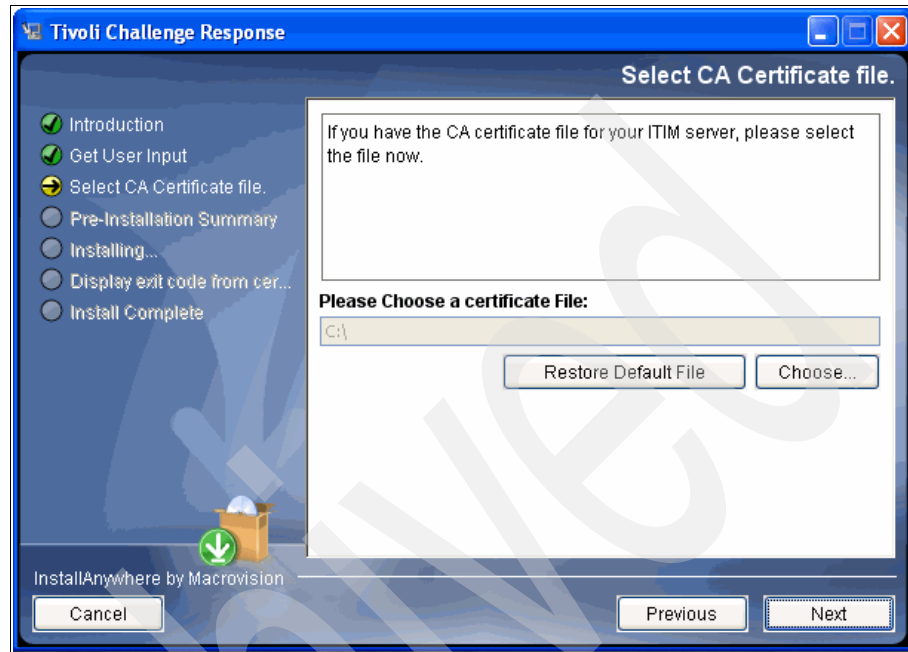


Figure B-2 Certificate file location

6. In the Install Summary window, review the installation settings. Click **Back** to change any of these settings. Otherwise, click **Next** to begin the installation.
7. In the Install Completed window, click **Finish** to exit the program.

The system will be restarted, and the next time the user logs on, the Tivoli Identity Manager Desktop Password Reset Assistant dialog is displayed automatically in the top left corner of the desktop with the Windows logon window. Every time the user tries to log on to the system, the DPRA is displayed as shown in Figure B-3 on page 637.

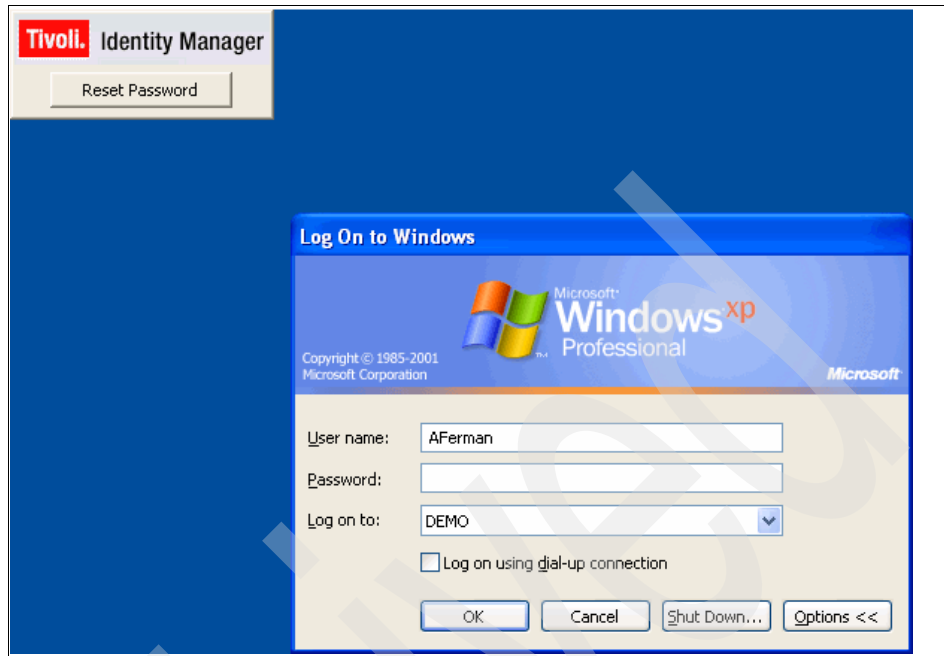


Figure B-3 Tivoli Identity Manager Desktop Password Reset Assistant displayed

At this point, the users are able to reset their passwords before gaining access to their desktop, or unlock their Windows account passwords.

Using DPRA

When users have forgotten their Windows password, or have locked their Windows account trying to access their desktops, they are now able to use DPRA to solve both problems.

The process flow is described in the next steps.

1. At the Windows logon interface, click the DPRA button for **Reset Password**. This will start the process.

2. Users have to authenticate to Tivoli Identity Manager by providing their user ID, as displayed in Figure B-4.

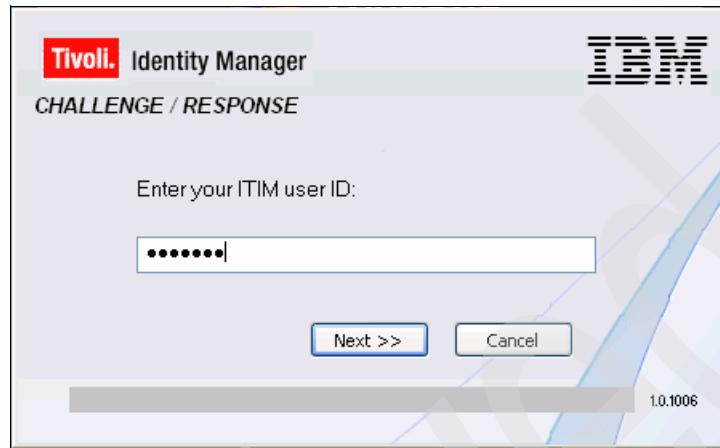


Figure B-4 DPRA user interface

3. The users are then challenged to provide the information stored in Tivoli Identity Manager (challenge/response feature).
4. The answers are validated against centrally stored Tivoli Identity Manager information. If the users provide the correct answers they will be allowed to reset their passwords, or to unlock their Windows account, as shown in next Figure B-5. In case the users do not provide the correct information an error message is displayed.

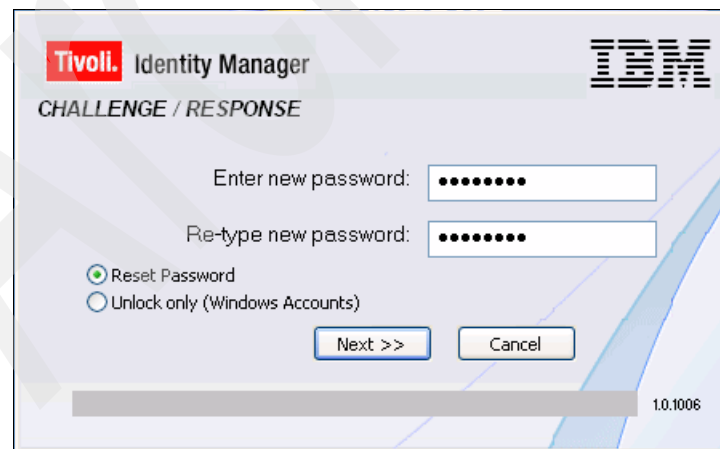


Figure B-5 Reset Windows account

This capability reduces calls to help desk by allowing users to reset their own password or unlock their Windows accounts even before they have access to their desktops.

Note: The behavior of the DPRA can be customized and changed, for detailed information about changing it, refer to *Tivoli Identity Manager Version 5.1 Desktop Password Reset Assistant Installation and User Guide, SC23-9625*.

DPRA customization

As described in 7.1.1, “Geographic distribution of TAA” on page 300, TAA is expanding its offices to Mexico City. For this reason, the DPRA user interface should be displayed in the Spanish language to users located in Mexico City.

DPRA provides the facility to modify the user interface. After installing the DPRA to desktops, TAA is now in the process to customize the user interface and change DPRA messages to TAA specific strings.

Note: The DPRA already has translated string tables for all of the supported languages. The language is selected based on the current code page.

The customization is intended to change the text of the supplied strings (and translations). This allows custom prompts and titles.

It is also possible to use custom graphics for the background bitmap and logo. For detailed information about changing it, refer to the *Tivoli Identity Manager Version 5.1 Desktop Password Reset Assistant Installation and User Guide, SC23-9625*.

You can replace the user interface labels by using the Desktop Password Reset Assistant. When you replace the labels, you must add the labels you want to override. The Desktop Password Reset Assistant searches for the updated labels at the following location:

File name	DPRA_Str.txt
File location	File location c:\windows\system32
Format	ID number,Newstring

The complete list of ID numbers and default strings is described in the DPRAstringIDs.txt file and is supplied for your reference. A sample DPRA_Str.txt is also supplied as a reference.

Note: The DPRA_Str.txt file must be a Unicode file.

The content of the file deployed in TAA's environment is shown in Example B-1.

Example B-1 DPRA_Str.txt file content

```
2019,OK
10005,Usuario de ITIM no encontrado
10006,Respuesta no valida
10007,La contraseña no cumple las reglas
20001,Introduzca su usuario de ITIM :
20003,Cargando preguntas de ITIM ...
20004,Validando respuestas con ITIM ...
20005,Introduzca su nueva contraseña:
20006,Verificacion de nueva contraseña:
20007,Error al validar las respuestas
20008,Enviando la peticion de sincronizacion de contraseña a ITIM ...
20009,Peticion de cambio de contraseña exitosa
20010,Fallo el cambio de constraseña
20011,Las contraseñas no son iguales
20012,Preguntas / Respuestas
20013,Pregunta %1!d! de %2!d!
20014,Siguiente >>
20015,<< Anterior
20016,Enviar
20017,Salir
20022,Unicamente desbloquear su cuenta de Windows
20023,Cambiar Contraseña
```

After deploying the file on the desktops located in Mexico City, those users can now use the DPRA in Spanish language, as shown in the next

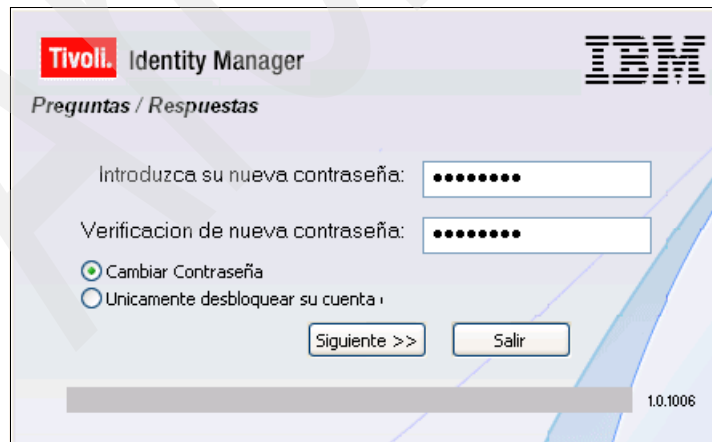


Figure B-6 DPRA labels customized

Note: The logo and background image of the DPRA can be customized and changed as well. For detailed information about changing it, refer to *Tivoli Identity Manager Version 5.1 Desktop Password Reset Assistant Installation and User Guide*, SC23-9625.

Conclusion

DPRA enables Windows users to perform self-service password resets and self-service account unlocks. Tivoli Identity Manager also provides a self-service user interface that allows users to manage their basic information, change their password, request for specific accesses, view their accounts, and perform activities related to identity management.

In combination with a Single Sign-On solution (like Tivoli Access Manager for Enterprise Single Sign-On), both solutions can deliver efficient user life cycle administration and can help to dramatically reduce calls to help desk, and reduce non-productive time caused by lost or missing access privileges.

Archived

Automating tasks for role management

In this appendix, we take a closer look at how to automate tasks for role management. We look at how *apiscript* can be used to help in this type of task.

Apiscript is a simple front-end to *wsadmin* (IBM WebSphere Application Server administrative scripting tool) which enables the use of the wsadmin Jython environment as a scripting environment for the Tivoli Identity Manager public API. The solution targets the following public APIs:

- ▶ Container management
- ▶ Person management
- ▶ Policy management
- ▶ Role management
- ▶ Request management

Requirements

Over the coming months TAA will be extending their RBAC implementation. This will involve defining greater numbers of roles and populating these roles with employee's accordingly. The data collected by the data gathering teams already reside in reusable formats such as spreadsheets and database tables. TAA require a means to reuse this data so roles and role membership can be bulk loaded quickly into Tivoli Identity Manager to reduce the risk of mistakes during manual data entry and to lessen the effort required.

Design considerations

The focus for this design is on static organizational roles, TAA implements what few dynamic organizational roles they have identified via the Tivoli Identity Manager administration interface. The design considers and addresses the following:

- ▶ Support input via comma separated value (CSV) text files
- ▶ Support the following role attributes:
 - Role name
 - Description
 - Target container
 - Role classification (optional)
 - Role owner (optional)
- ▶ Automate the population of the new roles with its initial membership

TAA's implementation

In this section, we detail the steps TAA follow to implement this solution.

Note: The files that are being used in this appendix are available via the additional material download. For more information refer to Appendix D, "Additional material" on page 655.

Installation and configuration of Apiscript

Apiscript is available for download from the IBM Tivoli Open Process Automation Library (OPAL, which contains integration examples for Tivoli Identity Manager and other Tivoli solutions).

<http://www.ibm.com/software/brandcatalog/portal/opal/details?NavCode=1TW10IM14>

The installation and configuration of apiscript is very simple and instructions are provided with the apiscript download. The TAA Identity Manager administrators download apiscript and install it onto their WebSphere Application Server following the instructions provided. If you want the convenience of not having to specify the full path when running apiscript you also need to add the apiscript *bin* directory to the systems PATH.

Note: We recommend you install apiscript into your Tivoli Identity Manager extensions directory.

```
<ITIM_HOME>/itim/extensions
```

Once installed, verify that apiscript is functioning correctly by running one of the example scripts provided in the `apiscript/examples` directory.

Run the example script using the following command:

```
apiscript.bat -f test_org.py
```

This example creates a new organization level container called *My Org*.

Note: The above example is designed for a Windows installation. For UNIX or Linux use the command `apiscript.ksh -f test_org.py`

Figure C-1 shows the administration interface Manage Organization Structure view of the newly created organization *My Org*, which was created by the apiscript `test_org.py`. Once you have verified it was created successfully it can be deleted.



Figure C-1 Manage organization structures

Bulk loading static organizational roles

TAA decided on the following CSV file format, which will contain the role data provided by the data gathering teams to be loaded into Tivoli Identity Manager.

Role name, description, org name, org type, role classification, role owner

Example (all on one line):

Baggage handling, Baggage handling role, AustinCSC, ou,
role.classification.business, hgold

The apiscript in Example C-1 is created to parse the data file and create the defined roles.

Example C-1 `bulkLoadRoles.py`

```
# import w/ rename for less typing
from apiscript.util import orgchart, orgrole, person
from java.util import Collections
from java.lang import String
import sys
```



```

# Use this script to bulk load static roles. This is useful when your planning or
data gathering already
# has or can easily provide the required data in CSV format.

# usage: apiscript.bat -f bulkLoadRoles.py [file name]

rolelist = sys.argv[0]

#Update to your organizations base org.
org_cont_base = orgchart.get_containers_by_attr("o","Tivoli Austin
Airlines",org_cont_mo=None, container_mgr=None).toArray()[0]

# Open file containing role list and read all lines.
# File format role name, role description, org name, org type, [role
classification],[role owner]
# Example file, note [role classification],[role owner] can be optional
# Role one,Description 1,AustinCSC,ou,role.classification.business,hgold
# Role two,Description 2,AustinCSC,ou,role.classification.business,
# Role three,Description 3,AustinCSC,ou,,
myfile = open(rolelist,"r")
line_mo = myfile.readlines()
myfile.close()

for line in line_mo:
    f = line.split(",")
    rolename=f[0].strip()
    desc=f[1].strip()
    org=f[2].strip()
    type=f[3].strip()
    classification=f[4].strip()
    owner=f[5].strip()

    if owner != "":
        print "Role owner as been specified so getting their person details"
        person_mgr=person.get_default_person_mgr()
        my_person=person_mgr.getPeople("uid",owner,org_cont_base,1).toArray()[0]
        print "Found person %s " % my_person.data.name
    else:
        print "No role owner specified, skipping person lookup"

    org_cont_mo = orgchart.get_containers_by_attr(type,org,org_cont_mo=None,
container_mgr=None).toArray()[0]
    print "Setting ITIM container for new role to %s" % org_cont_mo.data.name
    print "Creating role %s" % rolename
    data = {}

```

```

data["errolename"]=rolename
data["description"]=desc
data["erroleclassification"]=classification
if owner != "":
    data["owner"]=Collections.singletonList(my_person.distinguishedName.toString())
orgrole.submit_create_role_from_dict(org_cont_mo, data, t=None, role_mgr=None)
print ""

```

Running the bulkLoadRoles.py script

To run the script enter and run the following command:

```
apiscrpt.bat -f bulkLoadRoles.py rolelist.txt
```

The output of this command is depicted in Figure C-2.

```

C:\Program Files\IBM\itim\extensions\apiscrpt-5.0.0.4\apiscrpt\examples>apiscrpt.bat -f bulkLoadRoles.py rolelist.txt
Warning: using "default" for APISCRPT_ETC_HOSTNAME as "C:\Program Files\IBM\itim\extensions\apiscrpt-5.0.0.4\apiscrpt\etc\host\TIMFIVE.properties" does not exist.
Using host properties: "C:\Program Files\IBM\itim\extensions\apiscrpt-5.0.0.4\apiscrpt\etc\host\default.properties"
Using APISCRPT_WAS_HOME: "C:\Program Files\IBM\WebSphere\AppServer"
Using APISCRPT_ITIM_HOME: "C:\Program Files\IBM\itim"
WASX7357I: By request, this scripting client is not connected to any server process. Certain configuration and application operations will be available in local mode.
Welcome to IBM Tivoli Identity Manager API Scripting Tool (apiscrpt) version: 5.0.0.4
Setting system property: java.security.auth.login.config
Setting com.ibm.CORBA.properties: loginSource, loginUserId, loginPassword
WASX7303I: The following options are passed to the scripting environment and are available as arguments that are stored in the argv variable: [rolelist.txt]
Logging configuration file is not found. All the logging information will be sent to the console.
Role owner as been specified so getting their person details
Found person Pete P Drew
Setting ITIM container for new role to AustinCSC
Creating role Baggage Handling Role

Role owner as been specified so getting their person details
Found person Pete P Drew
Setting ITIM container for new role to AustinCSC
Creating role Baggage Handling Supervisors

Role owner as been specified so getting their person details
Found person Harry H Gold
Setting ITIM container for new role to AustinCSC
Creating role Check-in Desk Role

No role owner specified, skipping person lookup
Setting ITIM container for new role to AustinCSC
Creating role Check-in Desk Supervisors

No role owner specified, skipping person lookup
Setting ITIM container for new role to AustinCSC
Creating role Catering Airside Role

Role owner as been specified so getting their person details
Found person Harry H Gold
Setting ITIM container for new role to DenverCSC
Creating role Catering Airside Supervisors

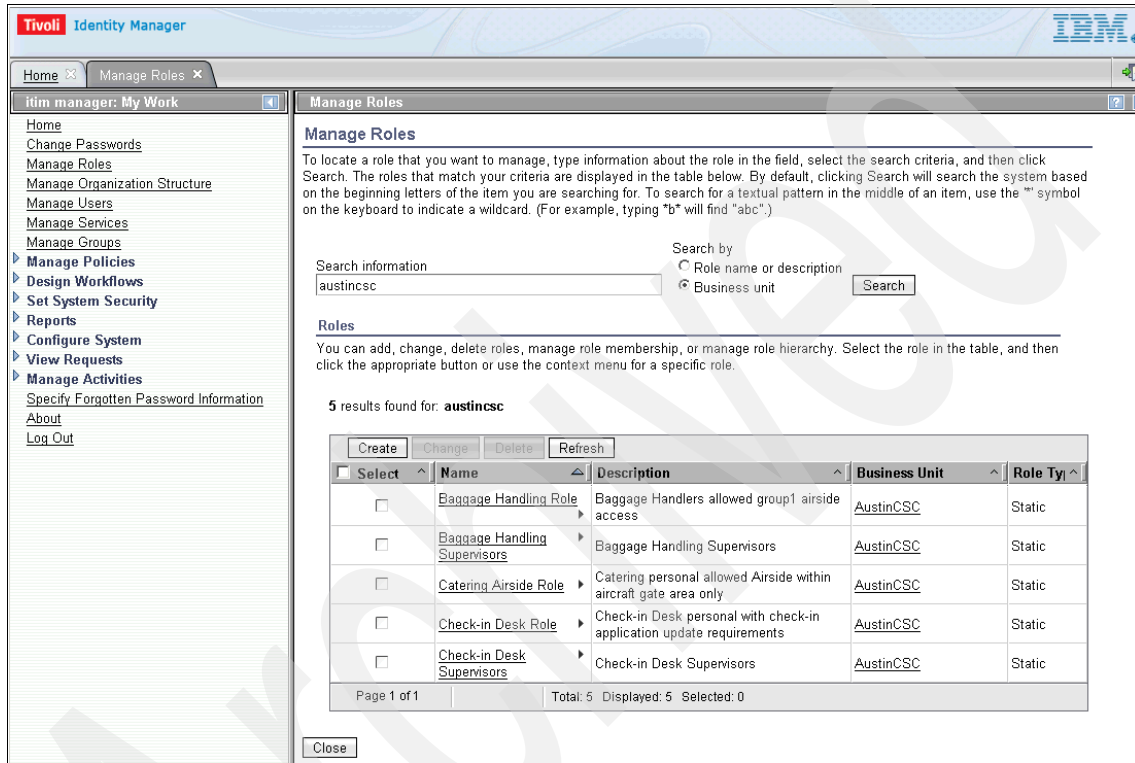
C:\Program Files\IBM\itim\extensions\apiscrpt-5.0.0.4\apiscrpt\examples>_

```

Figure C-2 Apiscrpt bulkLoadRoles.py

Verifying the newly created roles

Once the apiscript finishes execution, log in to the Tivoli Identity Manager administrative interface. Navigate to Manage Roles and verify a sample of the new roles to ensure they were created successfully. Enter *austincsc* into the search information box, then select Search by → Business unit and click **Search**. Figure C-3 shows the expected search results.



The screenshot shows the Tivoli Identity Manager administrative interface. The left sidebar contains a navigation menu with options like Home, Change Passwords, Manage Roles, Manage Organization Structure, Manage Users, Manage Services, Manage Groups, Manage Policies, Design Workflows, Set System Security, Reports, Configure System, View Requests, and Manage Activities. The main content area is titled 'Manage Roles' and includes a search section with a search information box containing 'austincsc' and radio buttons for 'Role name or description' and 'Business unit'. Below the search section is a table of search results for 'austincsc'.

Select	Name	Description	Business Unit	Role Type
<input type="checkbox"/>	Baggage Handling Role	Baggage Handlers allowed group1 airside access	AustinCSC	Static
<input type="checkbox"/>	Baggage Handling Supervisors	Baggage Handling Supervisors	AustinCSC	Static
<input type="checkbox"/>	Catering Airside Role	Catering personal allowed Airside within aircraft gate area only	AustinCSC	Static
<input type="checkbox"/>	Check-in Desk Role	Check-in Desk personal with check-in application update requirements	AustinCSC	Static
<input type="checkbox"/>	Check-in Desk Supervisors	Check-in Desk Supervisors	AustinCSC	Static

Figure C-3 Manage roles

Populating static organizational roles using apiscript

Once the new roles are created, TAA require that the initial role membership is also bulk loaded. This reduces the administration overhead by reusing the data that is already gathered. TAA decide that an apiscript script that takes two arguments will suffice. The first argument is the *role name* and the second argument is the *data file*. The file contains the employee's unique company id, one per line. This is known as the *uid*.

Example C-2 shows a sample data excerpt from the file.

Example: C-2 Input data file example for populating new roles

```
aparker
sfoulds
hgold
pdrew
sjenkins
thogg
thogg1
```

Example C-3 depicts the source for the apiscript that bulk loads the uid information into the new roles.

Example C-3 addPerson2Role.py

```
# import w/ rename for less typing
from apiscript.util import orgchart, person, orgrole, wait_requests
from com.ibm.itim.apps import Request
from java.lang import String
import sys

#
# Use this script to bulk load role members. This is useful when your planning or
data gathering already
# has or can easily provide files containing the role membership for each role.
# usage: apiscript.bat -f addPerson2Role.py [rolename] [data file]
# Data file should contain list of uid's of all new members of the role defined
# in rolename argument.

rolename = sys.argv[0]
filename = sys.argv[1]

# get the org under which to search
# edit line below to use another 'o' or 'ou'. Currently ou & AustinCSC
#
org_cont_mo = orgchart.get_containers_by_attr("ou","AustinCSC",org_cont_mo=None,
container_mgr=None).toArray()[0]
# org_cont_mo = orgchart.get_default_org_mo()
print "Searching for role in container %s" % org_cont_mo.data.name

# get the role to which to add the new members
role_mo = orgrole.do_get_roles_by_name(org_cont_mo, rolename).toArray()[0]
print "Found role %s " % role_mo.data.name
```

```

# Open file containing new role members and read all lines
myfile = open(filename,"r")
line_mo = myfile.readlines()
myfile.close()

# Using the line_mo list all members
# for each member define the person_mo, then add person_mo to
# the role previously defined in role_mo
org_cont_base = orgchart.get_containers_by_attr("o","Tivoli Austin
Airlines",org_cont_mo=None, container_mgr=None).toArray()[0]
#
for member in line_mo:
    p = member.strip()
    person_mgr=person.get_default_person_mgr()
    person_mo=person_mgr.getPeople("uid",p,org_cont_base,1).toArray()[0]
    print "Adding %s to role" % person_mo.data.name

    request = role_mo.addMember(person_mo, None)
    print "waiting on request %s" % request.ID
    wait_requests([request])
    assert request.status == Request.SUCCEEDED
    print "Done"
    print ""

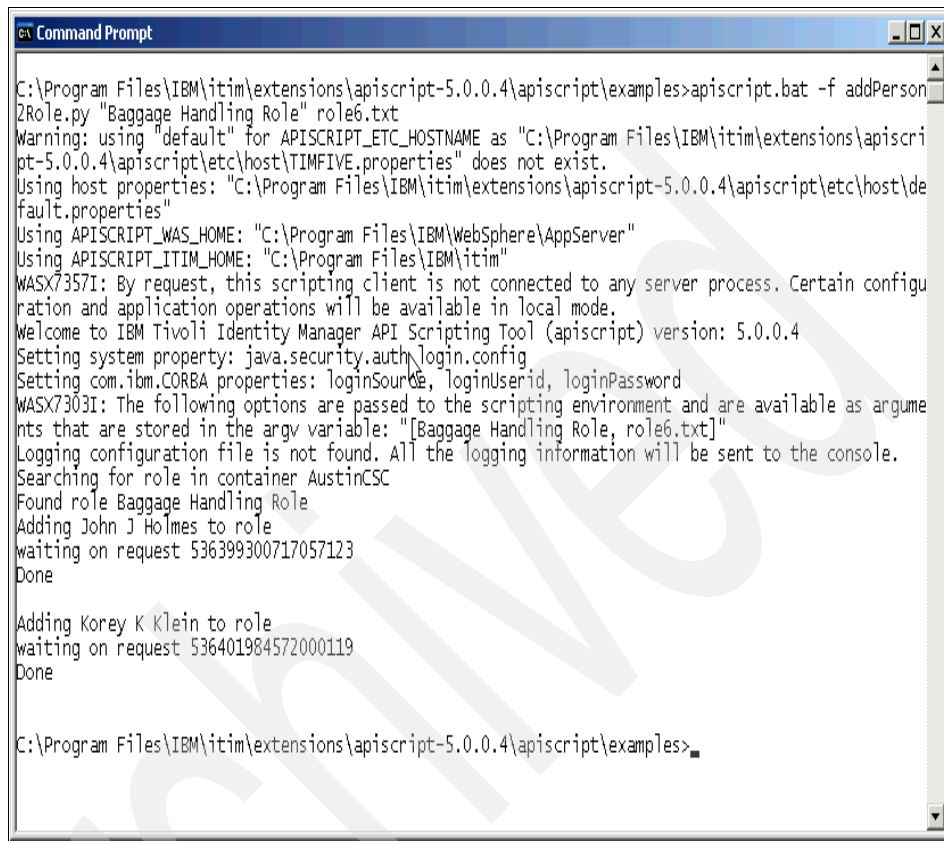
```

Running the addPerson2Role.py script

To run the script enter and run the following command:

```
apiscrypt.bat -f addPerson2Role.py "Baggage Handling Role" role6.txt
```

The output of this command is depicted in Figure C-4.



```
C:\Program Files\IBM\itim\extensions\apiscript-5.0.0.4\apiscript\examples>apiscript.bat -f addPerson2Role.py "Baggage Handling Role" role6.txt
Warning: using "default" for APISCRIPTETC_HOSTNAME as "C:\Program Files\IBM\itim\extensions\apiscript-5.0.0.4\apiscript\etc\host\TIMFIVE.properties" does not exist.
Using host properties: "C:\Program Files\IBM\itim\extensions\apiscript-5.0.0.4\apiscript\etc\host\default.properties"
Using APISCRIPTWAS_HOME: "C:\Program Files\IBM\WebSphere\AppServer"
Using APISCRIPITIM_HOME: "C:\Program Files\IBM\itim"
MASX7357I: By request, this scripting client is not connected to any server process. Certain configuration and application operations will be available in local mode.
Welcome to IBM Tivoli Identity Manager API Scripting Tool (apiscript) version: 5.0.0.4
Setting system property: java.security.auth.login.config
Setting com.ibm.CORBA.properties: loginSource, loginUserId, loginPassword
MASX7303I: The following options are passed to the scripting environment and are available as arguments that are stored in the argv variable: "[Baggage Handling Role, role6.txt]"
Logging configuration file is not found. All the logging information will be sent to the console.
Searching for role in container AustinCSC
Found role Baggage Handling Role
Adding John J Holmes to role
waiting on request 536399300717057123
Done

Adding Korey K Klein to role
waiting on request 536401984572000119
Done

C:\Program Files\IBM\itim\extensions\apiscript-5.0.0.4\apiscript\examples>
```

Figure C-4 Apiscript addPerson2Role.py

Verifying the new role membership

Once the apiscript finishes execution, log in to the Tivoli Identity Manager administrative interface. Navigate to Manage Roles and search for the role that was just updated. In our example this was the *Baggage Handling Role*. Enter the role name and click **Search**. The result is shown in Figure C-5.

The screenshot shows the Tivoli Identity Manager administrative interface. The left sidebar contains a navigation menu with options like Home, Change Passwords, Manage Roles, Manage Organization Structure, Manage Users, Manage Services, Manage Groups, Manage Policies, Design Workflows, Set System Security, Reports, Configure System, View Requests, Manage Activities, Specify Forgotten Password Information, About, and Log Out. The main content area is titled 'Manage Roles' and includes a search section with a search information field containing 'baggage handling role' and search criteria for 'Role name or description'. Below the search section, it states '1 results found for: baggage handling role'. A table displays the search results with columns for Select, Name, Description, Business Unit, and Role Type. The table contains one row for 'Baggage Handling Role' with description 'Baggage Handlers allowed group1 airside', business unit 'AustinCSC', and role type 'Static'. A context menu is open over the 'Baggage Handling Role' row, showing options: Change, Delete, Manage User Members (highlighted), Manage Child Roles..., Add User Members..., and Add Child Roles....

Select	Name	Description	Business Unit	Role Type
<input type="checkbox"/>	Baggage Handling Role	Baggage Handlers allowed group1 airside	AustinCSC	Static

Figure C-5 Manage Role: manage user members

Now click the twisty next to the role name *Baggage Handling Role* and select Manage User Members. This displays the Manage User Members and Child Roles dialog. Click **Search** to list all role members (Figure C-6).

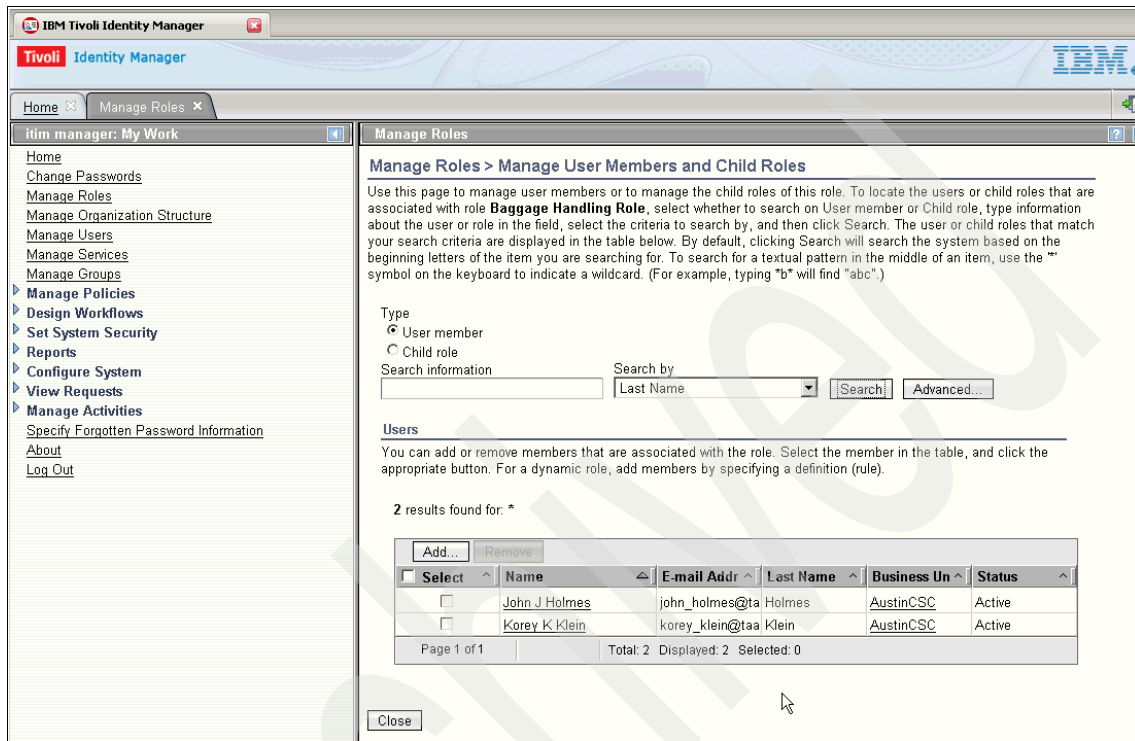


Figure C-6 Manage Role: showing current role members

Conclusion

In this appendix we demonstrated how using apiscript tasks can be automated and used for tasks in role management. The data collected during the Identity Manager planning and preparation stages was reused to aid the system administrator in deploying Tivoli Identity Manager quickly and accurately.

Apiscript can be used for many other administration tasks. We recommend you visit the IBM developerWorks Wiki for Tivoli Identity Manager to learn more and share your own experiences and scripts with others.

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

Additional material

This IBM Redbooks publication refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks publications Web server. Point your Web browser at:

<ftp://www.redbooks.ibm.com/redbooks/SG246996>

Alternatively, you can go to the IBM Redbooks publications Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks publication form number, SG24-6996.

Using the Web material

The additional Web material that accompanies this book includes the following files:

<i>File name</i>	<i>Description</i>
SG246996.zip	Tivoli Directory Integrator identity feed AssemblyLine

System requirements for downloading the Web material

In order for you to use the provided AssemblyLine you must have Tivoli Directory Integrator v6.1 or v7.0 installed on your machine.

How to use the Web material

Create a subdirectory (folder) on your workstation and unzip the contents of the Web material zip file into this folder.

From there you can open the identity feed AssemblyLine with your Tivoli Directory Integrator. the taaData folder also provides some sample data files.

The ApiScript directory contains the necessary files to follow our example in Appendix C, “Automating tasks for role management” on page 643.

Glossary

access (1) The ability to read, update, delete, or otherwise use a resource. Access to protected resources is usually controlled by system software. (2) The ability to use data that is stored and protected on a computer system.

access control In computer security, the process of ensuring that the resources of a computer system can be accessed only by principals in authorized ways.

access control item (ACI) Data that (a) identifies the permissions of principals and (b) is assigned to a resource.

access control list In computer security, a list that is associated with a resource that identifies all the principals that can access the resource and the permissions for those principals.

account An entity that contains a set of parameters that define the application-specific attributes of a principal, which include the identity, user profile, and credentials.

ACI target The resource for which you define the access control items. For example, an ACI target can be a service.

adapter (1) A set of software components that communicate with an integration broker and with applications or technologies in order to perform tasks, such as executing application logic or exchanging data. (2) A transparent, intermediary software component that allows different software components with different interfaces to work together.

administrative domain A logical collection of resources that is used to separate responsibilities and manage permissions.

adopt To assign an orphan account to the appropriate owner.

adoption rules. The set of rules that determine which orphan accounts belong to which owners.

agent A process that manages target resources on behalf of a system in order to respond to requests.

aggregate message A collection of notification messages that are combined into a single e-mail, along with optional user defined text.

alias In identity management, an identity for a user, which might match the user ID. The alias is used during reconciliation to determine who owns the account. A person can have several aliases, for example, GSmith, GWSmith, and SmithG.

approval A type of workflow activity that allows someone to approve or reject a request. See also workflow.

audit trail A chronological record of events or transactions. You can use audit trails for examining or reconstructing a sequence of events or transactions, managing security, and for recovering lost transactions.

authentication The process of verifying that an entity is the entity that it claims to be, often by verifying a user ID and password combination. Authentication does not identify the permissions that a person has in the system.

authorization The process of granting a user either complete or restricted access to an object, resource, or function.

Certificate Authority (CA) An organization that issues certificates. The CA authenticates the certificate owner's identity and the services that the owner is authorized to use, issues new certificates, renews existing certificates, and revokes certificates that belong to users who are no longer authorized to use them.

challenge-response authentication An authentication method that requires users to respond to a prompt by providing information to verify their identity when they log in to the system. For example, when users forget their password, they are prompted (challenged) with a question to which they must provide an answer (response) in order to either receive a new password or receive a hint for specifying the correct password.

Common Criteria A standardized method, which is used by international governments, the United States federal government, and other organizations, for expressing security requirements in order to assess the security and assurance of technology products.

connector A plug-in that is used to access and update data sources. A connector accesses the data and separates out the details of data manipulations and relationships.

credentials Authentication information that is associated with a principal.

CSV In computers, a CSV (comma-separated values) file contains the values in a table as a series of ASCII text lines organized so that each column value is separated by a comma from the next column's value and each row starts a new line. Here's an example:

```
Doe,John,944-7077
Johnson,Mary,370-3920
Smith,Abigail,299-3958
```

A CSV file is a way to collect the data from any table so that it can be conveyed as input to another table-oriented application. Spreadsheet programs or relational database applications can read CSV files. A CSV file is sometimes referred to as a flat file.

DAC Discretionary access control (DAC) is used to control access by restricting a subject's access to an object. It is generally used to limit a user's access to a file. In this type of access control, it is the owner of the file who controls other users' accesses to the file. Using a DAC mechanism allows users control over access rights to their files. When these rights are managed correctly, only those users specified by the owner may have some combination of read, write, execute, and so on, permissions to the file.

DAML Directory Access Markup Language. An XML specification that extends the functions of Directory Services Markup Language (DSML) 1.0 in order to represent directory operations. In Tivoli Identity Manager, DAML is mainly used for server to agent communications. See also Directory Services Markup Language v2.0.

Directory server A server that can add, delete, change, or search directory information about behalf of a client.

Directory Services Markup Language v1.0 (DSMLv1) An XML implementation that describes the structure of data in a directory and the state of the directory. DSML can be used to locate data into a directory. DSMLv1 is an open standard defined by OASIS. Contrast with Directory Services Markup Language v2.0.

Directory Services Markup Language v2.0 (DSMLv2) An XML implementation that describes the operations that a directory can perform (such as how to create, modify, and delete data) as well as the results of those operations. While DSMLv1 can be used to describe the structure of data in a directory, DSMLv2 can be used to communicate with other products about that data. DSMLv2 is an open standard defined by OASIS. Contrast with DSMLv1.

distinguished name (DN) The name that uniquely identifies an entry in a directory. A distinguished name is made up of name-component pairs. For example, CN=John Doe, O=My Organization, C=US.

domain administrator The owner of an administrative domain.

dynamic content tags A set of XML tags (based on the XML Text Template Language (XTTL) schema) that allows the administrator to provide customized information in a message, notification, or report.

dynamic organizational role An organizational role that is assigned to a person by using an LDAP filter. When a user is added to the system and the LDAP filter parameters are met, the user is automatically added to the dynamic organizational role.

entitlement In security management, a data structure, service, or list of attributes that contains externalized security policy information.

entitlement workflow A workflow that defines the business logic that is used when provisioning a policy. For example, an entitlement workflow is used to define approvals for managing accounts.

entity A person or object about which you want to store information or manage. For example, a person and an organization are both entities.

entity type Categories of managed objects. See also entity.

escalation The process that defines what happens and who acts when an activity has not been completed in the specified amount of time.

escalation limit The amount of time, for example, hours or days, that a participant must respond to a request, before an escalation occurs.

event The encapsulated data that is sent as a result of an occurrence, or situation, in the system.

failover An operation that switches a system to a redundant or standby system when services fail.

FESI extension A Java extension that can be used to enhance JavaScript code and then be embedded within a FESI script.

Free EcmaScript Interpreter (FESI) An implementation of the EcmaScript scripting language, which is an ISO standard scripting language that is similar to the JavaScript scripting language.

group A collection of Tivoli Identity Manager users.

identity The subset of profile data that uniquely represents a person or entity and that is stored in one or more repositories.

identity feed The automated process of creating one or more identities from one or more common sources of identity data.

identity policy The policy that defines the user ID to be used when creating an account for a user.

IIOB (Internet Inter-ORB Protocol) A protocol that is used for communication between Common Object Request Broker Architecture (CORBA) object request brokers (ORBs).

JDBC Java Database Connectivity is an application program interface (API) specification for connecting programs written in Java to the data in popular databases. The application program interface lets you encode access request statements in SQL that are then passed to the program that manages the database. It returns the results through a similar interface.

JMS Java Message Service is an application program interface from Sun Microsystems that supports the formal communication known as messaging between computers in a network. Sun's JMS provides a common interface to standard messaging protocols and also to special messaging services in support of Java programs. Sun advocates the use of the Java Message Service for anyone developing Java applications, which can be run from any major operating system platform.

JNDI Java Naming and Directory Interface enables Java platform-based applications to access multiple naming and directory services. Part of the Java Enterprise application programming interface (API) set, JNDI makes it possible for developers to create portable applications that are enabled for a number of different naming and directory services, including file systems, directory services, such as Lightweight Directory Access Protocol (LDAP), Novell Directory Services, and Network Information System (NIS), and distributed object systems, such as the Common Object Request Broker Architecture (CORBA), Java Remote Method Invocation (RMI), and Enterprise JavaBeans (EJB).

join directive The set of rules that define how to handle attributes when two or more provisioning policies are applied. Two or more policies might have overlapping scope, so the join directive specifies what actions to take when this overlap occurs.

JSP Java Server Page is a technology for controlling the content or appearance of Web pages through the use of servlets, which are small programs that are specified in the Web page and run on the Web server to modify the Web page before it is sent to the user who requested it.

Kerberos Kerberos is a secure method for authenticating a request for a service in a computer network. Kerberos was developed in the Athena Project at the Massachusetts Institute of Technology (MIT). The name is taken from Greek mythology; Kerberos was a three-headed dog who guarded the gates of Hades. Kerberos lets a user request an encrypted *ticket* from an authentication process that can then be used to request a particular service from a server. The user's password does not have to pass through the network.

LDAP Lightweight Directory Access Protocol is a software protocol for enabling anyone to locate organizations, individuals, and other resources, such as files and devices, in a network, whether on the public Internet or on a corporate intranet. LDAP is a *lightweight* (smaller amount of code) version of Directory Access Protocol (DAP), which is part of X.500, a standard for directory services in a network.

LDIF (LDAP Data Interchange Format) A file format that is used to describe directory information as well as changes that must be applied to a directory, such that directory information can be exchanged between directory servers that are using LDAP.

life cycle Passage or transformation through different stages over time. For example markets, brands, and offerings have life cycles.

life cycle rules A set of rules in a policy that determine which operations to use when automatically handling commonly occurring events, such as suspending an account that has been inactive for a period of time.

location An entity that is a subdivision of an organization, usually based on geographical area.

MAC The need for a mandatory access control (MAC) mechanism arises when the security policy of a system dictates that protection decisions must not be decided by the object owner and the system must enforce the protection decisions (for example, the system enforces the security policy over the wishes or intentions of the object owner). The POSIX.6 standard provides support for a mandatory access control policy by providing a labeling mechanism and a set of interfaces that can be used to determine access based on the MAC policy.

managed resource An entity that exists in the runtime environment of an IT system and that can be managed.

MASS IBM Method for Architecting Secure Solutions

operation An action that can be performed against an object; for example, add, modify, or delete.

operational workflow A workflow that defines the life cycle process for accounts, persons, and other entities.

organization A hierarchical arrangement of organizational units, such that each user is included once and only once.

organization tree A hierarchical structure of an organization that provides a logical place to create, access, and store organizational information.

organizational container An organization, organizational unit, location, business partner unit, or administration domain.

organizational role In identity management, a list of account owners that is used to determine which entitlements are provisioned to them.

organizational unit A type of organizational container that represents a department or similar grouping of people.

orphan account On a managed resource, an account whose owner cannot be automatically determined by the provisioning system.

password retrieval The method of retrieving a new or changed password by accessing a designated Web site and specifying a shared secret.

password strength policy A policy that defines the password strength rules. A password strength policy is applied whenever a password is set or modified.

password strength rules The set of rules that a password must conform to, such as the length of the password and the type of characters that are allowed (or not allowed) in the password.

password synchronization The process of coordinating passwords across services and systems such that only a single password is needed to access those multiple services and systems.

person An individual in the system that has a person record in one or more corporate directories.

post office A component that collects notifications from the appropriate workflow activities and distributes those notifications to the appropriate workflow participants.

provisioning The process of providing, deploying, and tracking a service or component.

provisioning policy A policy that defines the access to various managed resources, such as applications or operating systems. Access is granted to all users, users with a specific role, or users who are not members of a specific role.

RBAC. With RBAC (role-based access control), security is managed at a level that corresponds closely to the organization's structure. Each user is assigned one or more roles, and each role is assigned one or more privileges that are permitted to users in that role. Security administration with RBAC consists of determining the operations that must be executed by persons in particular jobs, and assigning employees to the proper roles. Complexities introduced by mutually exclusive roles or role hierarchies are handled by the RBAC software, making security administration easier.

reconciliation The process of synchronizing data in a central data repository with data on a managed resource.

request for information (RFI) A workflow activity that requests additional information from the specified participant.

ROI. For a given use of money in an enterprise, the ROI (return on investment) is how much *return*, usually profit or cost saving, results. An ROI calculation is sometimes used along with other approaches to develop a business case for a given proposal. The overall ROI for an enterprise is sometimes used as a way to grade how well a company is managed. If an enterprise has the immediate objectives of getting market revenue share, building infrastructure, positioning itself for sale, or other objectives, a return on investment might be measured in terms of meeting one or more of these objectives rather than in immediate profit or cost saving.

rule A set of conditional statements that enable computer systems to identify relationships and execute automated responses accordingly.

schema The fields and rules in a repository that comprise a profile.

scope In identity management, the set of entities that a policy or an access control item (ACI) can affect.

service A representation of a managed resource, application, database, or system.

service owner A role that identifies the person who owns and maintains a particular service in Tivoli Identity Manager. See also *service*.

service selection policy A policy that determines which service to use in a provisioning policy. See also *provisioning policy*.

service type A category of related services that share the same schemas. See also *service*.

SOAP Simple Object Access Protocol is a way for a program running in one kind of operating system to communicate with a program in the same or another kind of an operating system by using the HTTP Protocol and XML as the mechanisms for information exchange.

SSL The Secure Sockets Layer is a commonly-used protocol for managing the security of a message transmission on the Internet. SSL has recently been succeeded by Transport Layer Security (TLS), which is based on SSL.

static organizational role An organizational role that is manually assigned to a person. See also *organizational role*.

supervisor A role that identifies the person who supervises another set of users and who is often responsible for approving or rejecting requests that are made by those users.

suspend To deactivate an account so that the account owner cannot access the service.

system administrator A role that identifies the person who is responsible for the configuration, administration, and maintenance of Tivoli Identity Manager.

universally unique identifier (UUID) The 128-bit numerical identifier that is used to ensure that two entities do not have the same identifier. The identifier is unique for all space and time.

work order A workflow activity that requires a participant to perform an activity outside of the scope of the system.

workflow The sequence of activities performed in accordance with the business processes of an enterprise.

XML Extensible Markup Language is a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere. For example, computer makers might agree on a standard or common way to describe the information about a computer product (processor speed, memory size, and so forth) and then describe the product information format with XML. Such a standard way of describing data would enable a user to send an intelligent agent (a program) to each computer maker's Web site, gather data, and then make a valid comparison. XML can be used by any individual or group of individuals or companies that want to share information in a consistent way.

Archived

Related publications

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

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 667. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Business Process Reengineering and Beyond*, SG24-2590
- ▶ *Compliance Management Design Guide with IBM Tivoli Compliance Insight Manager*, SG24-7530
- ▶ *Continuous Business Process Management with HOLOSOFX BPM Suite and IBM MQSeries Workflow*, SG24-6590
- ▶ *Deployment Guide Series: IBM Tivoli Identity Manager 5.0*, SG24-6477
- ▶ *Deployment Guide Series: IBM Tivoli Access Manager for e-business V6.0*, SG24-7207
- ▶ *Enterprise Business Portals with IBM Tivoli Access Manager*, SG24-6556
- ▶ *Enterprise Business Portals II with IBM Tivoli Access Manager*, SG24-6885
- ▶ *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014
- ▶ *High Availability and Disaster Recovery Options for DB2 on Linux, UNIX, and Windows*, SG24-7363
- ▶ *IBM Tivoli Composite Application Manager Family Installation, Configuration, and Basic Usage*, SG24-7151
- ▶ *Intra-Enterprise Business Process Management*, SG24-6173
- ▶ *Problem Determination Using Self-Managing Autonomic Technology*, SG24-6665
- ▶ *Robust Data Synchronization with IBM Tivoli Directory Integrator*, SG24-6164
- ▶ *WebSphere Application Server V6 Planning and Design WebSphere Handbook Series*, SG24-6446
- ▶ *WebSphere Application Server V6 Scalability and Performance Handbook*, SG24-6392

- ▶ *WebSphere Application Server V6.1: System Management and Configuration*, SG24-7304

Other publications

These publications are also relevant as further information sources:

- ▶ Bass, et al, *Software Architecture in Practice, Second Edition*, Addison Wesley, 1997, ISBN 0321154959
- ▶ Committee on Information Systems Trustworthiness, et al, *Trust in Cyberspace*, National Academy Press, 1998, ISBN 0309065585
- ▶ Harris, *CISSP All-in-One Exam Guide*, The McGraw Hill Companies, 2001, ISBN 0072193530
- ▶ *IBM Tivoli Access Manager for e-business WebSEAL Administration Guide Version 6.1*, SC23-6505
- ▶ *IBM Tivoli Directory Integrator 6.1.1: Getting Started Guide*, GI11-6480-01
- ▶ *IBM Tivoli Directory Integrator 6.1.1: Reference Guide*, SC32-2566-01
- ▶ *IBM Tivoli Directory Integrator 6.1.1: Users Guide*, SC32-2568-01
- ▶ *IBM Tivoli Directory Integrator 7.0: Getting Started Guide*, GI11-8185
- ▶ *IBM Tivoli Directory Integrator 7.0: Reference Guide*, SC23-6562
- ▶ *IBM Tivoli Directory Integrator 7.0: Users Guide*, SC23-6561
- ▶ *IBM Tivoli Directory Server Administration Guide Version 6.1*, GC32-1564-00
- ▶ *IBM Tivoli Identity Manager 5.1 Troubleshooting and support*, available in the Tivoli Identity Manager Version 5.1 Information Center
- ▶ *IBM Tivoli Identity Manager Database and Directory Schema Reference Version 5.1*, SC27-2413
- ▶ *IBM Tivoli Identity Manager Planning Version 5.1*, available in the Tivoli Identity Manager Version 5.1 Information Center
- ▶ J. J. Whitmore, "A Method for designing Secure Solutions," IBM Systems Journal Vol. 40, No. 3, 747-768 (2001)
- ▶ Lloyd, et al, "Technical Reference Architectures," IBM Systems Journal Vol. 38, No. 1, 51-75 (1999)
- ▶ Rehtin, *Systems Architecting: Creating and Building Complex Systems*, Prentice Hall, Incorporated, 1990, ISBN 0138803455
- ▶ RFC2254 The String Representation of LDAP Search Filters

- ▶ *Tivoli Identity Manager: Tivoli Access Manager Combo Adapter Installation and Configuration Guide Version 5.1*, GC23-9664
- ▶ *Tivoli Identity Manager Password Synchronization for Active Directory Plug-in Installation and Configuration Guide Version 5.1*, SC23-9622
- ▶ *Tivoli Identity Manager Server Installation and Configuration Guide Version 5.1*, SC27-2410
- ▶ *Tivoli Identity Manager Version 5.0 & 5.1 Performance Tuning Guide*, SC23-6594
- ▶ *Tivoli Identity Manager Version 5.1 Desktop Password Reset Assistant Installation and User Guide*, SC23-9625

Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ National Institute of Standards and Technologies homepage
<http://www.nist.gov/>
- ▶ The Tivoli Identity Manager Version 5.1 Information Center:
<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itim.doc/welcome.htm>

How to get IBM Redbooks publications

You can search for, view, or download Redbooks publications, Redpapers publications, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks publications or CD-ROMs, at this Web site:

ibm.com/redbooks

Archived

Index

A

- access 106, 113, 127
 - defining ... 535
 - implementation 573
 - recertification 150
 - request workflow 122, 161
 - types 108
 - workflows 470
- access control 68, 70
 - ... for Identity Manager's keystore 272
 - ... to LDAP data 349, 353
 - delegation 477
 - HR database 371
 - management 59
 - model 14, 20
- Access Control Item
 - see ACI
- Access Manager 95
 - access control list 290
 - account management 276
 - adapter 276, 278
 - entitlement 283, 285, 288
 - groups 286
 - identity policy 279
 - integration with Identity Manager 275
 - Password Synchronization Adapter 277
 - pdadmin 282, 286
 - pkmspasswd 277, 287
 - Policy Server 97, 310, 348
 - provisioning policy 283
 - reconciliation 286
 - reverse password synchronization 282, 287
 - sec_master 287
 - server principals 287
 - service 278–279, 286
 - single sign-on 277
 - SSO with Identity Manager 294
 - SvrSslCfg 279
 - UNIX endpoint 282
 - Web Portal Manager 282, 286
 - WebSEAL 97, 291
 - WebSEAL load balancing 243
- Access Manager for Business Integration 309
- Access Manager for e-business 309
- Access Manager for Operating Systems 295
 - password management 282
- account 102–103, 127
 - certification process 578
 - compliance 406
 - compliance status 121
 - creation 337, 488, 611
 - creation of common account 421
 - defaults 118, 121, 128
 - entitlements 356
 - ID generation 147, 389
 - management 86, 90, 134, 140, 337, 456, 464, 611
 - Access Manager 276
 - role based 338
 - management delegation 337, 612
 - management historical data 330
 - management operations 326
 - manual management 110
 - migration of ... 233
 - non-compliant 338, 358, 360, 363, 612
 - orphan 103, 119, 150, 200
 - owner 406
 - password policy 431
 - password strength policy 433
 - provisioning 113, 141, 161
 - provisioning for Access Manager 288
 - RBAC relations 524
 - recertification 122, 129, 150, 167
 - reconciliation 158
 - removal 331
 - request workflow 122, 161
 - suspend operation 374
 - suspension 122, 337, 443, 611
 - system 235
 - user ID generation 113
 - workflow 470
 - workflow customization 615
- account group
 - access 536
- accountability 414
- ACI 86, 116–117, 188, 376, 462
 - placement 477

- principal 463
- action item 177
- Active Directory 35
 - adapter 347
 - identity feed 369
- activity 166
- adapter 108
 - Access Manager 276, 278
 - communication 109
 - connectivity 92, 216
 - customization 213, 608
 - high availability 256
 - monitoring 265
 - profile for Access Manager 279
 - security 351, 355
 - security configuration 272
 - server 347
 - SSL authentication 354
 - SSL communication 272
 - type 421
- add workflow 390
- admin domain 111, 289, 422, 462
- administration 422
 - ... of services 358
- administrative costs 327
- administrator
 - group 289
 - manual account management 110
 - role 288
- adoption 286, 406
 - ... of system accounts 235
 - policy 119
 - rule 128, 182, 407
- adoption policy 150
- advanced provisioning parameters 285
- agentCfg 352
 - password 273
- AIXProfile 296
- alert non-compliance 142
- alias 150, 421
- All role 424
- anonymity 70
- anonymous access
 - ... for LDAP 271
- apiscript 643
 - download 645
- application
 - client 209
 - configuration 228
 - extension 168
 - extensions for workflows 210
 - monitoring 262
- Application Layer 85
- application role 527
- application server
 - high availability 242
- approval 11, 338, 478
 - node 167, 201
 - process 327
 - workflow 118, 481
- approval of roles 115
- architectural decision 74–75
- archival
 - ... of data 237
- AssemblyLine 217, 293
 - identity feed 394
- asynchronous messaging 90
- attribute 102
 - mapping 383
- audit 4, 6, 11, 59, 68, 123, 161, 170, 319, 325, 331, 338, 405, 612
 - archival of data 237
 - data integrity 333
 - data migration 233
 - records archival 239
 - trail 92
- auditor group 289
- authentication 70, 333
 - module 89
 - monitoring 263
- authoritative data source 61
- authority
 - delegation 134
- authorization
 - module 90
- automatic
 - adoption 286
 - provisioning 143, 276
- automatic for Access Manager
 - provisioning 288
- automation
 - ... for role management 643
 - business process 13
- availability 241
 - adapter 265

B

- backup 334, 520
 - strategy 240
- Basel II 4
- best practices 319
- BIRT 170
 - reporting 606
- bulk loading roles 646
- business
 - continuity 260
 - entitlements 277
 - unit 127
- Business Intelligence and Reporting Tools
 - see BIRT
- business partner organization 111
- business process 51, 161, 166, 232
 - automation 13
 - delegation 470
 - flow 70
 - re-engineering 80
 - requirements 481
- business requirements
 - identity management foundation 324
- business role 527

C

- centralized user management 8
- certificate 279
- certification 331
 - process 578
- challenge/response 105, 132, 134–135, 277, 338, 452, 612
- change
 - control 503
 - management 334
- child role 182
- cluster deployment 221
- cn=anybody 353
- comma separated file
 - see CSV file
- common account
 - creation 421
- Common Criteria 68
- communication
 - adapter to server 109
- compliance 4, 6, 161, 170, 325, 329, 331, 405
 - alert 177, 478, 494
 - archival of data 237

- SOX 337, 611
- component
 - architecture 76
 - design 101
 - placement 94
- Composite Application Monitoring 269
- conceptual structure 53
- confidentiality 332
- configuration 343
 - ... of ports 99
 - management 227
 - migration of ... 234
 - settings 228
 - user interface 131
- conflict of interest 566
- connectivity 92
- connector 217
- contract
 - expiration 163
- controlled network 94
- CORBA 660
- corporate security policy 324, 328
- correct non-compliance 142, 363
- cost savings 324, 327
- credential 310
 - life cycle 70
- cryptographic 70
- Crystal Reports 170
 - reporting 606
- CSV file
 - identity feed 368
- custom
 - adapter 213
 - report 170
- custom person 102, 371
 - class 213
 - creation 385
 - versioning 228
- customization 208
 - user interface 214
- CustomLabels.properties 387

D

- DAC
 - see Discretionary Access Control
- DAML 92, 658
 - based adapter 108
 - protocol security 352

- service provider 218
- data
 - confidentiality 332
 - integrity 181
 - management zones 313
 - mapping 64
 - model 61
 - reconciliation 159
 - services API 211
 - services module 91
 - synchronization 207
- Data Warehouse 240
- database 92
 - audit table archival 239
 - high availability 254
 - server 347
 - SSL communication 271
- DB2 227
 - high availability 254
 - High Availability Disaster Recovery 255
 - mutual takeover multiple partition 255
- default policy 14
- delegate 180
- delegated administration 48, 288, 422
 - Access Manager 276
- delegation
 - ... of administration 9, 461
 - ... of role management 553
 - access control 477
 - account management 337, 612
 - business process 470
 - policy management 471
 - service management 471
- delete account
 - workflow customization 615
- dependencies
 - ... on import/export 182
- deployment
 - ... cluster 221
 - ... for single server 221
 - ... of multiple environments 229
- de-provisioning
 - of accounts 141
- design 101
 - objectives 334
 - process 66
- Desktop Password Reset Assistant 631
- development environment 229
- difference evaluation 183
- Directory Integrator 212–213, 217, 227, 240, 347
 - Access Manager integration 276
 - AssemblyLine 293
 - dispatcher service 347
 - identity feed 369, 393
 - RMI interface 92
- Directory Server 227
 - high availability 249
 - load balancing 253
- directory strategy 35
- discovery process 158
- Discretionary Access Control 21, 513
- dispatcher service 347
- DMZ 94
- domain 288
 - administrator 289
- draft provisioning policy 146
- DSML
 - identity feed 369
- DSMLv2
 - service provider 217
- due care 6
- due diligence 6
- dynamic business entitlements 292
- dynamic role 86, 113

E

- EJB 660
- e-mail
 - notification 173
- employee status 384
- end node 166
- enforcement definition 494
- entities 102
- entitlement 142, 278, 296, 423, 525, 562
 - ... for Access Manager 283
 - Access Manager 288
 - conflict 528
 - mapping to provisioning policies and roles 530
 - workflow 162, 167
- entity
 - life cycle rule 204
 - relationships 127
- entity type 162, 400
 - life cycle rule 204
- ePerson object 102
- erAccountItem 464
- erGlobalId 377

- erLocale 384
- erPersonItem 464
- erPersonStatus 374, 384, 388
- erRole attribute 376
- erSupervisor attribute 376
- escalation 167
 - limit 200
- event
 - handler 217
 - notification 354
- exception 329
- excludeAccounts 287
- export 182
 - exceptions 233
- Extensible Markup Language 663
- external notification 173

F

- failure recovery 241
- fault tolerance 242
- file permissions 351, 354
 - settings 271
- filtered reconciliation 159
- firewall
 - port configuration 99
- flow control 68
- forgot your password 138
- form
 - design 387
- functional
 - design 76
 - person 235
 - requirements 325, 419

G

- global
 - adoption rule 407
 - life cycle rule 203
- global level
 - policy enforcement 143
- global policy enforcement 364
- government environment 21
- grace login 282
- Gramm-Leach-Bliley 4
- group 29, 182, 289
 - access 536
 - access implementation 576
 - accesses 107

- management 59, 105, 185, 338, 495
- permission 188
- recertification 598
- reconciliation 105
- group management 15
- GUI server 95

H

- HACMP 242
- hardware requirements 220
- Health Insurance Privacy and Accountability Act 5
- helpdesk assistant group 289
- hierarchy
 - ... of roles 45
- high availability 241, 334
- High Availability Disaster Recovery for DB2 255
- High Availability Manager 245
- HIPAA 5
- historical
 - information 92
 - reporting 206
- hooked reports 606
- HpxProfile 296
- HR feed
 - SSL communication 272
- HTTPS 93
- human resources management system 63
- hybrid provisioning model 45

I

- IBM DB2 Universal Database
 - see DB2
- IBM Method for Architecting Secure Solutions 66
- IBM Tivoli Compliance Insight Manager
 - see Tivoli Compliance Insight Manager
- IBM Tivoli Composite Application Monitoring
 - see Composite Application Monitoring
- IBM Tivoli Data Warehouse
 - see Data Warehouse
- IBM Tivoli Monitoring 268
- identification 70
- identity
 - management 86
 - module 86
 - policy 87, 128
- identity and credentials 68
- identity feed 212, 336, 347, 363, 611
 - design 368

- Directory Integrator 393
 - initial 368
 - initial load 380, 401
 - initiation 372
 - input data 370, 380
 - modify request 378
 - monitoring 266
 - output data 371
 - push/pull 373
 - SSL 354
 - Identity Governance Framework 5
 - identity management 39
 - Identity Manager Groups 289
 - identity policy 119, 147, 182, 278, 285, 296
 - Access Manager 279
 - IDI Data Feed 372
 - service 397
 - import 183
 - Access Manager data 276
 - exceptions 233
 - import/export 65, 181, 229, 232, 504
 - additional considerations 184
 - archival of data 238
 - exceptions 233
 - incremental data synchronizer 207
 - InetOrgPerson
 - identity feed 369
 - inetOrgPerson 102, 213, 384
 - Information Technology Infrastructure Library
 - see ITIL
 - infrastructure
 - monitoring 262
 - initial identity feed 368, 380
 - installation 343
 - log 123
 - integrity 351
 - inter process communication 243
 - interface
 - customization 214
 - IT best practices 319
 - ITIL 224
- J**
- J2EE
 - application monitoring 263
 - JMS 243
 - JAAS 86
 - Java
 - API 209, 211
 - Java Database Connectivity 660
 - Java Messaging Service
 - see JMS
 - Java Naming and Directory Interface 660
 - Java Server Pages 660
 - JavaScript 166
 - ... in Access Manager integration 283
 - ... in add workflow 391
 - ... in Directory Integrator identity feed 395
 - ... in identity feed 375
 - ... in identity policy 148, 280
 - ... in provisioning policy 208
 - ... in service selection policy 149, 422
 - ... in workflow 168
 - extensions 208, 607
 - JDBC 660
 - JMS 90, 660
 - high availability 243
 - Identity Manager reliability 246
 - local queue 246
 - reliability levels 244
 - shared queue 247
 - JNDI 660
 - service provider 397
 - job function 61
 - job role 113, 143, 306
 - join operations 425
 - JSP 660
- K**
- Kerberos 660
 - keystore
 - access control 272
- L**
- language 383
 - LDAP 660
 - anonymous access 271
 - automated failover 251
 - cn=anybody 353
 - custom person 385
 - data access control 349, 353
 - directory 91
 - ePerson object 102
 - excludeAccounts 287
 - group membership 286
 - high availability 249

- inetOrgPerson 102, 213, 384
- load balancing 253
- organizationalUnit 376
- replica 250
- replica server 310
- schema migration 234
- schema settings 228
- server 347
- SSL communication 271
- tag value pairs 291
- LDIF file 286
- leave of absence 384
- Liberty
 - Identity Governance Framework 5
- life cycle
 - management 5, 16, 161, 578
 - management module 87
 - monitoring 264
 - operation 384
 - rule 19, 48, 129, 163, 182, 203
 - rule workflow 122
- Lightweight Directory Access Protocol 660
- local queue 246
- location 111
- lock to-do list item 179
- logging 123, 330
 - module 91
- logical component architecture 84
- logical component design
 - service layer 89
- logical structure 53
- login policy 281
- loop 167, 200
- Lotus Notes
 - adapter 347

M

- MAC
 - see Mandatory Access Control
- mail
 - module 91
- maintenance
 - test and production environment 506
- manage people 138
- managed resource 103, 140, 358
- managed service 118
- management entities 110
- management objects
 - archival of data 238
 - management of accounts 140
 - manager
 - attribute 376
 - group 289
 - view 491
 - Mandatory Access Control 21, 513
 - manual account management 110
 - manual activity 168
 - notification 173
 - manual permissions 106
 - manual service 360
 - mapping
 - person attributes 383
 - mark non-compliance 142, 161
 - MASS 54
 - access control 68
 - architectural decision 74
 - audit 68
 - component architecture 76
 - flow control 68
 - functional design 76
 - identity and credentials 68
 - solution integrity 68
 - solution model 75
 - use case 75
 - membership 423
 - message queue
 - default setup 247
 - messaging
 - module 90
 - Meta Directory 35, 37
 - Method for Architecting Secure Solutions 54, 66
 - methodology 54
 - migration 338, 612
 - ... between environments 232
 - ... of data between environments 182
 - ... of test configuration 503
 - considerations 64
 - military environment 21
 - modify request 378
 - monitoring 260, 520
 - reconciliation 159
 - multiple environments 229

N

- naming context 398
- network diagram 307

- network zone 94, 347
 - controlled 94
 - restricted 95
 - secure 95
 - uncontrolled 94
- non-compliant 142, 161
 - account 331, 338, 358, 360, 363, 494, 612
- non-functional
 - requirements 335, 344
- notification template 174

O

- one-way password synchronization 104
- operation report 206
- operation workflow 122, 162–163, 167, 470
- orchestration
 - module 90
- organization 111
 - chart 184
- organization tree 102, 139, 355
 - design 356
 - migration of structural changes 233
 - service placement 359–360
- organizational role 113, 142, 288, 291, 296, 523, 525
 - access 535
 - accesses 107
 - import 183
- organizational unit 111
- organizationalPerson 213
- organizationalUnit 376
- orphan account 103, 119, 159–160, 200, 236, 337, 406, 611
 - adoption 150
 - cleanup 414
- Others role 424
- ownership
 - ... of roles 526

P

- parent relationship 375
- participant 166
- password
 - ... for directory and database server 351
 - challenge/response 452
 - change for Access Manager 277
 - change interceptor 337, 611
 - custom rules 211

- encryption for properties file 271
- expiration 163
- forgotten 138
- generator 211
- login policy 281
- management 9, 59, 103, 134, 326, 632
- policy 19, 87, 119, 128, 148, 182, 278, 431, 437
- policy for Access Manager 281
- policy for Windows 134
- reset 132, 306, 314, 326, 493
- rule API 148
- rule verification 442
- scheduled change 197
- self-reset 338, 612
- strength 113, 148
- strength checking 141
- strength policy 281, 433
- synchronization 103–104, 134, 281, 337, 436, 438, 611
 - Windows reset 631
- password synchronization
 - ... for Tivoli Access Manager 104
- Payment Card Industry Data Security Security Standard 4
- PCI 4
- PCI DSS 4
- pdadmin 282, 286
- pending requests 134, 199
- people
 - management 464
- performance 334
 - analysis 270
 - monitoring 540
 - tuning 237
- permission 188
- permission settings
 - ... for files 271
- permissions 106
- person 102
 - alias 150
 - archival of data 237
 - attribute mapping 383
 - custom class 213
 - custom person creation 385
 - delegate 180
 - entity 280
 - entity type 386
 - functional 235
 - management 138

- migration of ... 233
- object 127, 292
- preferred ID 150
- record data selection 371
- relationships 375
- status 388
- supervisor 389
- suspend operation 374
- personnel management 305
- physical
 - architecture 345
 - component architecture 94
 - structure 53
- pkmspasswd 277, 287
- placement
 - ... of components 94
 - rule 375, 383
- planning
 - multiple environments 229
- policy 113, 566
 - adoption 119
 - default 14
 - enforcement 8, 331
 - enforcement definition 494
 - evaluation 406
 - exception 329
 - global policy enforcement 364
 - identity 119, 147, 182, 285, 296
 - join behavior 528
 - management 86, 471
 - management module 87
 - module 89
 - non-compliance 142
 - password 119, 148, 182, 431, 437
 - password strength 433
 - preview 118
 - provisioning 118, 142, 162, 182, 285, 288, 291, 296, 363, 423, 458, 484, 562
 - provisioning policy 366
 - recertification 17, 120, 150, 229, 511
 - separation of duty 120, 154, 228, 511, 514, 536, 566
 - service selection 118, 149, 562
 - validation 14
- Policy Server 97
- port
 - configuration 99
 - number 279
- post office 175, 204
- preferred ID 150
- preferred language 383
- presentation layer 84
- preview provisioning policy 146
- principal 116, 463
- priority 423
 - join operation 425
- process 166
- production environment 229
- project team 55
- provisioning 90, 113, 288
 - ... of accounts 141
 - advanced parameters 285
 - automation 143
 - draft policy 146
 - engine 49
 - hybrid model 45
 - layer 84
 - management module 87
 - models 43
 - non-compliance 142
 - organizational policy 143
 - parameters 423
 - policy
 - mapping to entitlements and roles 530
 - policy 87, 113, 118, 142, 162, 182, 285, 288, 291, 296, 358, 363, 366, 423, 458, 484, 562
 - Access Manager 276
 - designing ... 528
 - entitlements 128
 - RBAC relations 524
 - policy for Access Manager 283
 - policy import 183
 - policy join operations 425
 - policy migration 232
 - policy preview 146
 - policy simulation 118
 - request-based 44
 - role-based 44
 - scope 146
 - services 146
 - workflow 162
- pseudonymity 70
- push/pull identity feed 373

Q

- quality
 - ... of service 319

- assurance 64
- quarantine container 375

R

- RACF 348
- random password 104
- RBAC 5, 14, 20, 662
 - components 514
 - model 26
 - organizational role 524
 - preparing for ... 514
 - provisioning policy 143
 - role definition 521
 - role design 30
 - role mapping 518
 - role ownership 526
 - scenario implementation 511
 - system design 30
- recertification 17, 129, 167, 331
 - ... for roles 116
 - policy 17, 120, 129, 150, 229, 511
 - policy implementation 579
 - scheduling of ... 202
 - workflow 122, 162
- reconciliation 93, 105, 158, 337, 363, 404, 421, 525, 611
 - ... of system accounts 236
 - Access Manager 276, 285–286
 - adoption 150
 - adoption policy 119
 - filter 236
 - IDI Data Feed 373
 - management module 87
 - password change 277
 - schedule 199, 234, 405, 408
- recovery 241, 334, 520
- recycle bin
 - archival of data 239
- Redbooks Web site 667
 - Contact us xvi
- relational database 92
- relationship 375
 - role-group 29
- reliability level 244
- Remote Method Invocation
 - see RMI
- remote services
 - module 91, 216
- report
 - data synchronization 207
 - definition versioning 229
 - migration of definition 233
- reporting 11, 48, 124, 206, 413, 448
 - advanced topics 606
 - BIRT 606
 - Crystal Reports 606
 - hooked reports 606
 - overview 13
 - system 170
- request
 - access 288
 - account 288
- request for information 167
- request-based provisioning 44, 535
- resource 140
 - access lists 290
 - management 290
- respond to to-do list item 179
- restricted network 95
- return on investment 5, 317, 662
- reverse password synchronization 134, 282, 287, 337, 348, 611
 - monitoring 264
- reverse proxy 310
- RFI 167, 170
- risk
 - assessment 7
 - management 318
 - mitigation 5
- RMI 92
 - based adapter 108
 - dispatcher service 347
 - service provider 218
- ROI 662
- role 29, 86, 113, 127, 143, 338, 424, 613
 - access 535
 - access implementation 575
 - administration
 - ACI 557
 - customization 562
 - All 424
 - approval 115, 548
 - workflow extension 548
 - bulk loading 646
 - changes 330
 - classification 522, 527
 - definition 61, 521

- hierarchy 45, 72, 89, 113, 339, 514, 521
- management 45, 59
 - automation 643
 - delegation 553
- mapping to provisioning policies and entitlements 530
- migration 232
- module 89
- Others 424
- ownership 526
- RBAC relations 524
- recertification 116, 598
- relationship 89, 113, 376, 514, 521
- sample mapping 518
- separation of duty policy 120
- Role Based Access Control 8
 - see RBAC
- role-based provisioning 44
- rollback 232

S

- Sample recertification policies 18
- Sarbanes-Oxley 4
- schedule 196
- schedule information 92
- scheduling
 - ... of changes 197
 - ... of reconciliations 199
 - module 90
- scope 423
- search management module 87
- sec_master 287
- secrets 70
- secure network 95
- Secure Sockets Layer 662
- security
 - architecture 51, 72
 - compliance 325
 - configuration 271
 - domain 288
 - policy 6, 59, 96, 290, 324, 328, 356–357
 - for Access Manager 277
 - policy exception 329
 - risk 332
 - role 290
 - SSL communications 350, 354
- security design objectives 334
- self-care 10

- user interface 131
- self-registration 209
- self-service 209
 - interfaces 48
- senior administrator 289
- sensitivity silo 25
- separation of duty 47, 72, 338, 566
 - exemptions 121, 158
 - policy 120, 154, 228, 511, 514, 536
 - policy exemption 330
 - prerequisites 523
 - violation 155
- separation of duty policy
 - delegation 572
- server
 - log 123
 - principals 287
 - sizing 348
 - statistics 270
- service 103, 110, 127, 182
 - ... selection policy 87, 118, 149
 - Access Manager 276, 278–279
 - account defaults 121
 - design 422
 - IDI Data Feed 373, 397
 - implementation 358
 - layer 89
 - level
 - policy enforcement 143
 - management 471
 - manual service 360
 - naming context 398
 - owner group 289
 - physical location 358
 - placement 422, 426
 - placement in org tree 359–360
 - profile 109, 184, 278, 421
 - profile adoption rule 407
 - profile installation 185
 - TDI Data Feed 382
 - type 110
 - account defaults 121
- Service Integration Bus 243
- service level agreement 260, 324
- service provider 216
 - JNDI 397
- service selection policy 182, 359, 422, 478, 480, 562
- shared queue 247

- Simple Object Access Protocol 662
- simulation 118
- single server deployment 221
- single sign-on 277
 - ... for Identity Manager 294
- sizing
 - of servers 348
- SOAP 662
- software
 - architecture 52
 - requirements 219
 - version control 227
- SolarisProfile 296
- solution
 - architecture 74
 - design 52
 - integrity 68
 - model 75
- SSL 98, 662
 - adapter communication 109
 - authentication for adapters 354
 - certificate 279
 - communication design 271
 - event notification 354
 - secure communications 350, 354
- start node 166
- Statement of Work 57
- static role 113
- status change 90
- strategic business initiative 318
- strength policy 281
- subform 215
- subject 280
- subprocess node 168
- supervisor 389
 - relationship 127, 376, 401
- support administrator 289
- suspend 138
 - operation 374
- suspend non-compliance 142
- suspension of account 337, 443, 611
- SvrSslCfg 279
- synchronization
 - ... of passwords 103, 281
 - Windows password 436, 438
- Synthetic Transaction Investigator 270
- system
 - account 235
 - administrator 117
 - management module 88
 - notification 173
 - systems integration 64
 - systems management zone 313

T

- tag/value support 292
- target
 - systems 59
 - type profiles 296
- TDI Data Feed
 - service 382
- test environment 181, 229, 504
 - maintenance 506
- Tivoli Compliance Insight Manager 6
- Tivoli Systems Automation
 - DB2 high availability 255
- to-do list
 - delegate 180
- transactional information 92
- transactions 70
- transition 166
- troubleshooting
 - import/export 184

U

- uncontrolled network 94
- union join operation 425
- UNIX
 - endpoint 282
 - user ID synchronization 295
- unlock to-do list item 179
- use case 75
- user 102
 - centralized management 8
 - group management 15
 - ID generation 113
 - permission 188
 - provisioning 35, 38
 - registry
 - Access Manager 276
- user interface 131
 - customization 214
 - subform 215
- user management 13, 58, 290
 - historical data 330
 - scenario 314
- user suspension

- scheduling of ... 198
- V**
- validation policy 14
 - version control 227
 - view 195
 - Virtual Meta Directory 37
- W**
- Web Portal Manager 282, 286
 - Web single sign-on
 - ... for Identity Manager 294
 - WebSEAL 95, 97, 282, 291, 309
 - business entitlements 277
 - dynamic business entitlements 292
 - load balancing 243
 - reverse password synchronization 282, 287
 - SSO with Identity Manager 294
 - tag/value support 292
 - WebSphere Application Server 221, 227
 - administrative scripting tool 643
 - High Availability Manager 245
 - Service Integration Bus 243
 - WebSphere Network Deployment Manager 221
 - Windows
 - desktop password reset 631
 - password policy 134
 - password reset 134
 - work order 168, 170, 177
 - workflow 11, 122, 161, 481
 - activity 166
 - add person 390
 - advanced customization 607
 - application extension 168
 - application extensions 210
 - approval 118
 - customization 210, 615
 - role approval 115
 - design 182
 - design interface 165
 - elements 166
 - engine 48
 - escalation limit 200
 - extension for rolw approval 548
 - interfaces 164
 - loop 200
 - management module 88
 - migration 232
 - module 90
 - operation 168
 - provisioning policy entitlement 528
 - recertification 583
 - recertification policy 120
 - requirements 169
 - script 168
 - time limits 200
 - user interface 169
 - versioning 228
 - wizard 164
 - WorldWide Project Management Methodology 54
 - wsadmin 643
 - WWPMM 54
- X**
- X.500 35, 660
 - X.509 certificate authentication 89
 - XML 92, 663

Archived



Redbooks

Identity Management Design Guide with IBM Tivoli Identity Manager



Identity Management Design Guide

with IBM Tivoli Identity Manager



Redbooks®

**Enterprise
integration for
identity life cycle
management**

**Complete
architecture and
component
discussion**

**IBM Tivoli Access
Manager integration**

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 it 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 an approach for designing an identity management solution with IBM Tivoli Identity Manager Version 5.1. Starting from the high-level, organizational viewpoint, we show how to define user registration and maintenance processes using the self-registration and self-care interfaces as well as the delegated administration capabilities. Using the integrated workflow, we automate the submission/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 wish to understand and implement a centralized identity management and security infrastructure.

**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**