IBM® Tivoli®

Enterprise Single Sign-On Design Guide

Using IBM Security Access Manager for Enterprise Single Sign-On 8.2

- Holistic design approach for an enterprise single sign-on project
- Complete information about architecture and components
- Real-world scenario with hands-on details

Axel Buecker
Nilesh Patel
Dirk Rahnenfuehrer
Joris Van Herzele

Redbooks

ibm.com/redbooks

International Technical Support Organization

Enterprise Single Sign-On Design Guide Using IBM Security Access Manager for Enterprise Single Sign-On 8.2

September 2012

SG24-7350-02

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

Third Edition (September 2012)

This edition applies to Version 8.2 of IBM Security Access Manager for Enterprise Single Sign-On.

Contents

Notices ix
Trademarks x

Preface xi
The team that wrote this book xi
Now you can become a published author, too! xiv
Comments welcome xiv
Stay connected to IBM Redbooks xv

Part 1. Architecture and design 1

Chapter 1. Business context 3
1.1 The single sign-on paradigm 4
1.2 Enterprise single sign-on today 5
1.2.1 Solving the password security paradox 5
1.2.2 Managing passwords in a security-rich fashion 6
1.2.3 Reducing help desk costs and improving employee productivity 6
1.2.4 Demonstrating compliance through auditing and reporting 7
1.2.5 Easy to deploy 7
1.2.6 High performance 7
1.2.7 Integrating with an enterprise identity management system 8
1.2.8 Bringing single sign-on to kiosk machines and virtual desktops 8
1.3 Considerations for deployment 9

Chapter 2. Single sign-on architecture and component design 11
2.1 Overview 12
2.1.1 IBM Security Blueprint perspective 13
2.2 Logical component architecture 16
2.2.1 AccessAgent 17
2.2.2 AccessAgent in server mode 33
2.2.3 IMS Server 33
2.2.4 IMS database 36
2.2.5 AccessAdmin 37
2.2.6 AccessStudio 39
2.2.7 Provisioning API 40
2.3 Additional components 41
2.4 Physical component architecture 42
2.4.1 IMS Server 43
2.4.2 IMS database 44

2.4.3 Organization directories 44
2.4.4 AccessAgent 46
2.5 IBM Security Access Manager for Enterprise Single Sign-On integration. 47
2.5.1 User provisioning products 47
2.5.2 Compliance products 48
2.5.3 Software provisioning products 48
2.5.4 Web single sign-on 48
2.5.5 User repositories 48
2.5.6 Database servers 48
2.5.7 Reporting tools 48
2.5.8 Monitoring products. 49
2.5.9 Third-party readers 49
2.5.10 Epic Electronic Health Records. 49
2.6 Conclusion. 49

Chapter 3. Solution design and management 51
3.1 Business requirements 52
3.1.1 Increasing security 52
3.1.2 Reducing costs and improving productivity. 54
3.1.3 Addressing compliance 55
3.2 Functional requirements 56
3.2.1 Comparing the various session management models 56
3.2.2 Operational security requirements 61
3.2.3 High-availability design 66
3.2.4 Multiple factor authentication 71
3.3 Deployment strategies. 75
3.3.1 Plan for IMS Server scalability 76
3.3.2 Deployment time estimation 78
3.3.3 Initial deployment scenario 78
3.3.4 Managing expectations 80
3.3.5 Enabling single sign-on for applications 81
3.4 Log collection and audit reporting 83
3.4.1 Audit log collection 83
3.4.2 Audit reporting. 84
3.5 Conclusion. 85

Part 2. Customer environment. 87

Chapter 4. Overview of scenario, requirements, and approach 89
4.1 Company overview 90
4.1.1 Current IT infrastructure 91
4.1.2 Security and usability issues within the current infrastructure. 95
4.2 Business vision 96

4.3 Business requirements 97
4.3.1 IBM Security Framework mapping to business requirements 98
4.4 Functional requirements 99
4.4.1 IBM Security Blueprint mapping to functional requirements 100
4.5 Design approach 103
4.6 Implementation approach 105
4.7 Conclusion 105

Chapter 5. Base installation and configuration 107
5.1 Design considerations 108
5.1.1 System requirements 109
5.1.2 Deployment architecture 109
5.2 Installing and configuring base components 111
5.2.1 Creating administrative users 111
5.2.2 Deploying the IMS Server Virtual Appliance 112
5.2.3 Starting the Virtual Appliance 117
5.2.4 Configuring the database server 128
5.2.5 Initial IMS Server configuration 129
5.2.6 Provisioning an IMS administrator and verifying the installation . . . 144
5.2.7 Configuring user and machine policy templates 148
5.2.8 Deploying AccessAgent 149
5.2.9 Interacting with AccessAgent 158
5.2.10 Installing AccessStudio 164
5.3 Configuring AccessProfile 167
5.3.1 IBM Lotus Notes application 169
5.3.2 SAP application 176
5.4 Managing the deployed environment 195
5.4.1 Managing policies 196
5.4.2 Managing users 196
5.4.3 Logging 197
5.5 Conclusion 197

Chapter 6. Password self-services implementation 199
6.1 Business requirements 200
6.2 Password self-service architecture 200
6.3 Implementing password self-service 200
6.3.1 Setting up the self-service questions 201
6.3.2 Enabling the password self-service function 206
6.3.3 User enrollment interview 210
6.3.4 Executing a password reset 213
6.4 Conclusion 225

Chapter 7. Strong authentication using RFID 227

7.1 Configuring machine and user policy templates 229
7.1.1 Basic configuration by using the Setup assistant 229
7.1.2 Configuring the personal workstation and RFID 236
7.1.3 Configuring shared workstations, shared desktops, and RFID 241
7.1.4 Configuring details for the user policy template 248
7.2 Using RFID 254

Chapter 8. Roaming desktop implementation 259
8.1 Cardio healthcare requirements 260
8.2 Overview of the roaming desktop features 261
8.2.1 Component architecture overview. 261
8.2.2 Logging on manually to the VMware virtual desktop. 263
8.3 Cardio healthcare implementation. 265
8.3.1 Usage scenarios 268
8.4 Conclusion. 270

Chapter 9. Implementing operational requirements 271
9.1 Fixes 272
9.1.1 Finding fix levels 272
9.1.2 Obtaining fixes 273
9.1.3 Receiving fix notifications 274
9.2 Audit log maintenance. 274
9.3 Database maintenance 275
9.4 Cached Wallet maintenance 275
9.5 Backup and restore procedures 275
9.5.1 WebSphere Application Server profile 276
9.5.2 IMS database 282
9.5.3 IMS Server configuration. 287
9.6 Tivoli Common Reporting 299
9.7 Conclusion. 308

Part 3. Appendixes 309

Appendix A. Renewing the Secure Sockets Layer certificate used by the IBM HTTP Server. 311
Procedure to renew a certificate. 312

Appendix B. Advanced profiling 317
Background 319
HTML 319
HTML and JavaScript 320
JavaScript and DHTML. 321
HTML load sequence 321
Document complete event and the Observer. 322

The first document complete event 322
Subsequent document complete events 325
Signatures 325
The web-signature 325
The HTML signature 327
Auto-learn AccessProfile 329
Handling basic authentication 329
Frames and the web browser document object 330
Web page B navigates to another web page within Document 2 333
A child document web page causes navigation in the parent document . 334
Frames support in V8.1 344
Frames support in V8.2 345
Differences between Firefox and Internet Explorer AccessProfiles 345
Common issues 346
Slowness due to multiple auto-learn loads 346
No injection because an HTML element was not found 347
No injection and no HTML element not found error 351
Captured credentials are obfuscated when saved 352
Save action not firing. 355
Slow injection or clicking when using fire after some time trigger 357
No AccessProfile loading for a non-browser hosted web page. 358
Use case 359
Use case: Start a program from the command line. 359
Conclusion 368

Appendix C. Configuring strong authentication 369
Configuring authentication to use smart cards 371
Prerequisite environment 371
Testing smart card compatibility 372
Configuring the certificate authority. 374
Importing the CA root certificate to the HTTP Server truststore, part 1 . . 378
Importing the CA root certificate to the HTTP Server truststore, part 2 . . 381
Enabling two-way Secure Sockets Layer on the IBM HTTP Server 390
Creating IMS Server policies for smart card use. 392
Assigning the new template to the client workstation 395
Modifying the user default template to accept smart card authentication . 395
Issuing a certificate to a smart card 397
Registering a smart card to a user 405
Configuring authentication to use radio frequency identification cards 408
Prerequisite environment 408
Creating and assigning the RFID machine policy template 409
Creating an authentication code for the user 412
Registering the RFID card to the user. 414

Strong authentication by using biometrics 416
Finger biometrics-based authentication 416
Configuring authentication to use fingerprint recognition 425
Prerequisite environment 425
Setting up fingerprint authentication 426
Updating the user template 437
Enrolling the user fingerprint for authentication. 440
Configuring authentication for Mobile ActiveCode as a one-time password . 444
Prerequisite environment 444
Creating the messaging connector for email. 445
Configuring AccessAssistant for MAC second-factor authentication 452
Configuring the user account for MAC use 455
Logging on with MAC 459
Conclusion 461

Related publications 463
IBM Redbooks 463
Other publications 463
Online resources 464
How to get Redbooks 465
Help from IBM 465

Index 467

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

Preface

Everyone feels the pain of too many passwords to remember. Everyone can relate to the security exposure of weak passwords, chosen for convenience. And, everyone can relate to passwords placed in proximity to the workstation for a quick reminder. Unfortunately, that note can allow more than the intended user into the system and network. The average user today often has four or more passwords. And, security policies that focus on password complexity and password-change frequency can cause even more difficulty for users.

This IBM® Redbooks® publication introduces IBM Security Access Manager for Enterprise Single Sign-On 8.2, which provides single sign-on to many applications, without a lengthy and complex implementation effort. Whether you are deploying strong authentication, implementing an enterprise-wide identity management initiative, or simply focusing on the sign-on challenges of a specific group of users, this solution can deliver the efficiencies and security that come with a well-crafted and comprehensive single sign-on solution.

This book is a valuable resource for security officers, administrators, and architects who want to understand and implement an identity management solution in a medium-scale environment.

The team that wrote this book

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

Axel Buecker is a Certified Consulting Software IT Specialist at the ITSO, Austin Center. He writes extensively and teaches IBM classes worldwide about areas of software security architecture and network computing technologies. He has a degree in Computer Science from the University of Bremen, Germany. He has 25 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.

Nilesh Patel is a Technical Leader in the IBM Security Software Group. He is a solution advisor for IBM Security and Compliance Management Solutions. He has extensive experience in design and implementation of Identity and Access Management (IAM) solutions along with Security Intelligence, Analytics, and Compliance. He has published many technical papers within the IBM developer domain and customized integration modules on the IBM Open Process Automation Library for IAM and Security Intelligence products. He has delivered many technical webcasts to educate clients on the new features and integration of IBM Security products.

Dirk Rahnenfuehrer is a Systems Management Specialist at IBM. He has 12 years of experience in the IT industry and holds a degree in Physics from RWTH Aachen University. He has worked at IBM for 10 years. His areas of expertise include Systems Management and Security products, focusing on identity management since 2003 and single sign-on since 2005.

Joris Van Herzele is an IT Architect at Umicore (a global materials technology group) where he delivers technology research, architectural solution design, and sourcing analysis for global projects. He is a Certified Information Systems Security Professional with more than 12 years of experience in various security domains. Before his current role, Joris worked at IBM Managed Security Services where he provided research, design, and evaluation analysis. He also taught classes through IBM X-Force education services, and was a pre-sales engineer advocating the IBM Security portfolio in EMEA.

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

Abdul Baki, Matthew Boult, Kelvin Chin, Mohit Chugh, Matthew Duggan, Sey Gan, Brian Goldsmith, Amit Kumar Saini, Steve Lay, Archit Lohokare, Chee Meng Low, Jessilou Noelle Lontok Lawas, Stefan Mueller, Fajar Priyanto, Daryl Romano, Nandagopal Seshagiri, Vivek Shankar, Dave Silvestro, Grey Thrasher, and Peter Wolf
IBM

Edwin D'Hondt
Umicore SA
Broekstraat 31 rue du Marais
1000 Brussels, Belgium

Sven Gossel and Saroj Paudel
charismathics GmbH
47 Sendlinger St
Munich, Germany 80331

Sean Dyon, Myles Tillotson, and Dennis Wilcox
Bio-Key International
3349 Highway 138 Building D, Suite A
Wall, New Jersey, US 07719

Thanks to the authors of the second edition of this book, *Deployment Guide Series: IBM Tivoli Access Manager for Enterprise Single Sign-On*, published in June 2009:

Steve Lay, Dirk Rahnenfuehrer, and Frank Sommer

Thanks to the authors of the first edition of this book, *Deployment Guide Series: IBM Tivoli Access Manager for Enterprise Single Sign-On*, published in March 2007:

Jose Fermaintt and Norman Field

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

Part 1

Architecture and design

In this part, we describe the business context of the IBM Security Access Manager for Enterprise Single Sign-On. We then explain how to technically architect the overall solution into an existing environment and introduce the logical and physical components.

1

Business context

In this chapter, we describe the business context for IBM Security Access Manager for Enterprise Single Sign-On (ISAM ESSO). After a short description of the single sign-on paradigm, we describe the factors that influence why and how IBM Security Access Manager for Enterprise Single Sign-On can be implemented in a certain business context. Finally, we explain the challenges that you can expect to encounter when deploying an single sign-on solution in a large enterprise.

1.1 The single sign-on paradigm

While the purpose and benefits of single sign-on technologies are widely understood and accepted today, the many variations and applications of these technologies are increasingly confusing to the contemporary organization. Single sign-on, broadly, can be classified into three major types, *web single sign-on*, *federated single sign-on*, and *desktop single sign-on*, depending on the type of applications and use cases prevalent in the organization. Most organizations today have all three types of applications deployed in their IT environment. It is important for the organization to address all these use cases through a common, comprehensive, and integrated single sign-on framework. The IBM Identity and Access Management portfolio provides an organization with just that framework.

Web single sign-on provides single sign-on between a web security server, such as the Access Manager for e-business WebSEAL component, and back-end web applications. Web single sign-on can eliminate the need to log in twice when a client attempts to access a web resource on a server that requires authentication from its own user registry. *Federated single sign-on* provides single sign-on between trusted web applications by using separate user registries, often between an organization and its business partner.

Desktop single sign-on, alternatively referred to as *enterprise single sign-on*, allows for a seamless, transparent single sign-on experience to many more applications than merely web applications. Enterprise single sign-on provides the users with the ability to log on with a single password to *existing and thick client applications,* such as email clients, Java applications, host and mainframe applications, custom business applications, and more.

From the perspective of the users, IBM Security Access Manager for Enterprise Single Sign-On is an agent that runs on their Microsoft Windows desktop. Users must remember a single password only to log on to any protected application on the desktop. IBM Security Access Manager for Enterprise Single Sign-On securely stores the login information for each application and responds automatically to any credential challenges posed by the application. It also provides centralized policy management and backup of user credentials.

Combining IBM Security Access Manager for Enterprise Single Sign-On and Tivoli Access Manager for e-business with a comprehensive identity management strategy allows companies to greatly reduce maintenance costs and security risks.

1.2 Enterprise single sign-on today

The number of logins and passwords that employees must manage on a daily basis continues to be a source of frustration and lost productivity. Employees must remember login information for numerous applications, many of which require different user names and passwords, different password complexity requirements, and forced password changes in ever shorter intervals. The number of logins an employee must manage grows with the deployment of each additional business application. The corporate help desk often bears the brunt of restoring lost or forgotten login information for an employee. These factors together contribute to security risks and increase help desk costs that few organizations can afford not to address.

So how can IBM Security Access Manager for Enterprise Single Sign-On benefit your organization? In this section, we describe how IBM Security Access Manager for Enterprise Single Sign-On addresses these serious security, productivity, and compliance challenges in a centrally managed solution.

1.2.1 Solving the password security paradox

A frequent user complaint is the requirement to remember multiple passwords, and a major security weakness in computer security is weak password selection. Security breaches as a result of weak passwords or insecure management of passwords are common. Allowing weak passwords can create security threats, such as hackers that are able to easily guess passwords by using *dictionary attacks* or by using *brute force* scripts. Forcing users to remember several complex strong passwords can also create security threats, such as users that write their passwords on a note or save them in a text file, because they have so many passwords that frequently change and are not easily remembered. The paradox is clear. You must enforce strong passwords to eliminate threats created by weak passwords, but you must also provide a mechanism so users do not have to manage multiple complex passwords.

IBM Security Access Manager for Enterprise Single Sign-On can help you solve this paradox. *Users must remember only one password* and they no longer must deal with strong passwords for all corporate applications. They authenticate once, and IBM Security Access Manager for Enterprise Single Sign-On does the rest.

In most cases, this solution is implemented without having to modify any of the business applications, including their password requirements. Additionally, IBM Security Access Manager for Enterprise Single Sign-On allows for increased security through *second-factor authentication* mechanisms. Radio Frequency Identification (RFID), building access badges, smart cards, and Short Message

Service (SMS) messaging are used, in addition to the user password. IBM Security Access Manager for Enterprise Single Sign-On also supports the use of finger biometrics for *strong authentication*. With this option enabled, no password is required for single sign-on, increasing security and further reducing help desk support costs. Unlike some of the other authentication methods mentioned, fingerprints cannot be shared, so the “user” authenticated is the “person” authorized. And with finger biometrics, there are no cards or other credentials that must be purchased, distributed, or replaced when lost.

IBM Security Access Manager for Enterprise Single Sign-On detects and responds to all password-related events to automate every password management task for the user, including login, password selection, password change, and password reset. IBM Security Access Manager for Enterprise Single Sign-On delivers single sign-on for Windows, web, Java, UNIX, command line, in-house developed applications, host-based mainframe applications, and custom business applications.

1.2.2 Managing passwords in a security-rich fashion

IBM Security Access Manager for Enterprise Single Sign-On secures password-related applications, and data. It uses the strongest cryptography currently available, including Advanced Encryption Standard (AES) and Triple Data Encryption Standard (DES). The IBM Security Access Manager for Enterprise Single Sign-On Federal Information Processing Standard (FIPS) 140-2 compliance can help financial institutions, government agencies, healthcare, and other organizations to comply with the stringent privacy and security regulations that govern their operations.

IBM Security Access Manager for Enterprise Single Sign-On also offers second-factor authentication to further increase security. It allows mixing and matching different factors, depending on the user or machine. If these factors exist, IBM Security Access Manager for Enterprise Single Sign-On can use them.

1.2.3 Reducing help desk costs and improving employee productivity

The IBM Security Access Manager for Enterprise Single Sign-On self-service password reset functionality can reduce or eliminate the costs associated with forgotten passwords and lost employee productivity due to account lockouts. Forgotten passwords and account lockouts, as a result of too many failed attempts, can burden the company help desk. IBM Security Access Manager for Enterprise Single Sign-On provides configurable functions to allow users to perform password self-service in ways that meet various security requirements.

How users interact with password changes, resets, and account lockout and unlock functions can be customized and allowed or disallowed based on configurable policies. IBM Security Access Manager for Enterprise Single Sign-On grants companies the flexibility to decide whether these password and account service functions stay with the help desk, the user, or a combination.

The password reset function provides the capability to reset your IBM Security Access Manager for Enterprise Single Sign-On password to regain access to your desktop environment. It does not reset any application-specific passwords.

1.2.4 Demonstrating compliance through auditing and reporting

IBM Security Access Manager for Enterprise Single Sign-On includes built-in auditing and reporting for fine-grained user activities on the enterprise desktop. It can record audit events, including user login and logout of applications. The audit mechanism can be customized to capture other relevant information related to user activities. The product ships with several included reports, but custom reports can be generated because all audit data resides in a single relational database that can be queried.

1.2.5 Easy to deploy

Implementing and managing IBM Security Access Manager for Enterprise Single Sign-On is made effective with a web-based administrative console, superior directory integration, and easily deployable client-side software. All administrative functions are performed from a centralized web administrative console (AccessAdmin). Point and click wizards in the AccessStudio application walk an administrator through the tasks of profile configuration. An administrator can access the AccessAdmin console from anywhere a web browser is able to connect to the server. IBM Security Access Manager for Enterprise Single Sign-On uses a pre-existing user repository without the need to modify the directory schema or any other aspect of the user repository.

1.2.6 High performance

In all private, shared, and roaming desktop environments[1], IBM Security Access Manager for Enterprise Single Sign-On can deliver uncompromising speed. It uses minimal resources when providing a single sign-on experience for users to their applications. With its event-specific resource usage, the impact of IBM Security Access Manager for Enterprise Single Sign-On on both the client and the network is minimal. No additional hardware or software is required.

[1] For more information about private, shared, and roaming desktop environments, see “Session management” on page 31.

1.2.7 Integrating with an enterprise identity management system

The IBM Security Access Manager for Enterprise Single Sign-On *Provisioning Bridge* extends the benefits generated by IBM Security Access Manager for Enterprise Single Sign-On through the automation of the credential distribution process. The IBM Security Access Manager for Enterprise Single Sign-On Provisioning Bridge uses its API libraries to allow identity management software to automatically provision IBM Security Access Manager for Enterprise Single Sign-On user credentials. This way, users never have to know their user name or password for their applications because it can be managed transparently to them.

If users need to know their user name and password for a particular application, they are able to obtain that information by accessing the credential store (*Wallet*). This access is possible only if they are authenticated to IBM Security Access Manager for Enterprise Single Sign-On. If they are not working at a workstation with an AccessAgent, they can access that information by using the AccessAssistant web browser-based interface. Even if not integrated with identity management software, IBM Security Access Manager for Enterprise Single Sign-On allows for a highly available and secure password-reveal process through these components.

1.2.8 Bringing single sign-on to kiosk machines and virtual desktops

The convenience of allowing others to share a workstation unfortunately does not come without risks. Too often users walk away from a kiosk[2] machine without logging off, potentially exposing sensitive data. IBM Security Access Manager for Enterprise Single Sign-On addresses this threat by its ability to automate the termination of inactive sessions and application shutdown. That automation includes features, such as automatic *walk away* logouts through RFID proximity keys, or smart card removal.

IBM Security Access Manager for Enterprise Single Sign-On provides robust session management support for *roaming desktop* implementations. It uses technologies, such as Windows Terminal Services and Citrix XenApps, and *shared desktops* or *kiosk machines,* as well as *private desktops*. Users can roam easily and securely from one workstation to another. IBM Security Access Manager for Enterprise Single Sign-On also includes support for securing Virtual Desktop Infrastructure (VDI) technologies, such as VMware View. Additionally, IBM Security Access Manager for Enterprise Single Sign-On includes thin client and client-less access through roaming desktop mode and through *Web Workplace*.

[2] Kiosk machines are also referred to as *shared* or *roaming* desktops. For more information about shared and roaming desktop environments, see “Session management” on page 31.

1.3 Considerations for deployment

Although IBM Security Access Manager for Enterprise Single Sign-On provides user-friendly tools for deploying and managing the product, there are many factors to consider before embarking on the deployment. As with any deployment, a planning document must be written that outlines the scope, the phases, and the time line of the deployment. At a minimum, the planning document must include the following factors:

- Deployment approach

  Because IBM Security Access Manager for Enterprise Single Sign-On involves deploying software to user workstations, the deployment must be carefully planned to avoid adversely affecting users. The rollout to the enterprise must be phased, starting by using IBM Security Access Manager for Enterprise Single Sign-On to secure the common desktop applications and then gradually adding new capabilities, such as other custom applications.

- Distribution of the software to the desktops

  IBM Security Access Manager for Enterprise Single Sign-On is a Windows-based application referred to as *AccessAgent*. The AccessAgent can be deployed as a Windows Installer (previously known as a Microsoft Installer) or MSI package that uses most standard software distribution technologies. AccessAgent communicates with the IBM Integrated Management System (IMS) Server[3] to synchronize data changes with the server. However, AccessAgent can cache data locally (on disk) based on policy. Therefore, AccessAgent can perform most of its functions in offline mode, even if it is not connected to the IMS Server at any point.

  Another aspect to consider is user education. After AccessAgent is distributed to the desktop, the user is initially prompted to *sign up* to use IBM Security Access Manager for Enterprise Single Sign-On. This task can be a forced action, but it can be configured so that the user has the choice to sign up. The uninformed user might be confused about how to decide. Sometimes, users also must interact with the client software if they want IBM Security Access Manager for Enterprise Single Sign-On to manage their password for a new application. A good practice is to announce upcoming deployments well in advance and to offer a simplified step-by-step help guide to support users through the sign-up process.

- Corporate security policies

[3] The IBM Security Access Manager for Enterprise Single Sign-On Integrated Management System Server (IMS Server) is one of the central components of the overall solution. For more information, see Chapter 2, “Single sign-on architecture and component design” on page 11.

Some configuration parameters in IBM Security Access Manager for Enterprise Single Sign-On are likely governed by corporate security policies. The administrative configuration console (*AccessAdmin*) provides the flexible configuration of security policies for users, machines, applications, and authentication mechanisms. You must work closely with security officers to ensure that the policy is correctly represented in the solution.

- Password reset strategy

  The ability for users to reset their own password on their own desktop machine is a powerful tool for the enterprise. However, care must be taken when formulating the series of questions that users must answer to reset their password. Issues of privacy and cultural sensitivity must be taken into account. You must review your security questions with your legal department before proceeding with your deployment to ensure that the questions comply with local privacy laws.

In the following chapter, we explain the overall architecture of IBM Security Access Manager for Enterprise Single Sign-On including the logical and physical components.

2

Single sign-on architecture and component design

In this chapter, we describe the logical and physical architecture of IBM Security Access Manager for Enterprise Single Sign-On and its fundamental components. First, we provide a general overview of the product. Next, we introduce the logical components within IBM Security Access Manager for Enterprise Single Sign-On and how they relate to each other. In the physical architecture section, we describe where to deploy the components that make up IBM Security Access Manager for Enterprise Single Sign-On. In the last section, we describe IBM Security Access Manager for Enterprise Single Sign-On integration with other products.

2.1 Overview

IBM Security Access Manager for Enterprise Single Sign-On provides its single sign-on functionality by introducing a layer that authenticates a user one time and then automatically detects and handles subsequent requests for user credentials. Figure 2-1 depicts an overview of the solution.

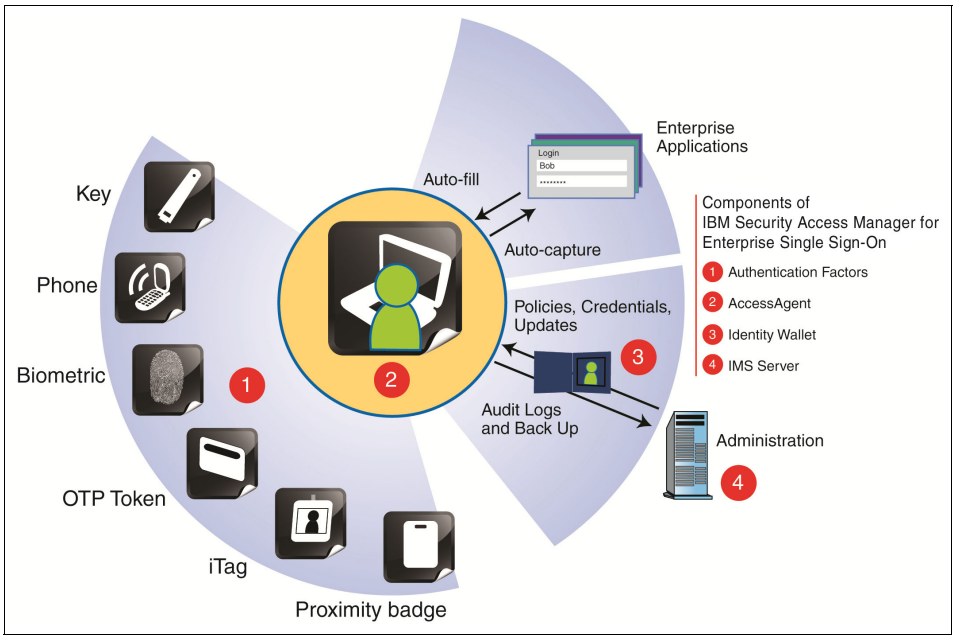

Figure 2-1 Product overview

IBM Security Access Manager for Enterprise Single Sign-On can be divided into the following functions:

- Authentication factors

  IBM Security Access Manager for Enterprise Single Sign-On supports various *authentication factors* to authenticate the user. Besides the standard user name/password-based authentication, the user can be authenticated by a proximity or building badge. Examples are active or passive radio frequency identification (RFID), a fingerprint, a one-time password (OTP) provided by Short Message Service (SMS) or OTP token, or a USB token.

- AccessAgent

  AccessAgent runs on every Windows desktop endpoint, Microsoft Windows Server Terminal Services session, VMware vSphere virtual desktop interface session, and Citrix XenApp Presentation Server session. AccessAgent is

responsible for authenticating the user[1]. It can automate single sign-on into Windows and to the set of applications that are defined in *AccessProfiles*. AccessAgent can extend the Windows Graphical Identification and Authentication (GINA) dynamic link library (DLL) chain to provide additional functions for self-service or strong authentication.

- Identity Wallet

  The *Identity Wallet* (or *Wallet*) holds the user credentials that are required for single sign-on. It is loaded from the IMS Server into AccessAgent after successful authentication of the user so that it is available even when the endpoint is disconnected from the computer network. To protect the credentials against tampering or stealing, the Identity Wallet is encrypted with a strong encryption mechanism.

- IMS Server

  The *Integrated Management System Server* (IMS Server) is the central repository for user data, AccessProfiles, Identity Wallets, and machine profiles. The IMS Server provides a web-based interface to administer users and policies.

2.1.1 IBM Security Blueprint perspective

IBM Security Access Manager for Enterprise Single Sign-On combines single sign-on, strong authentication, session management, access workflow automation, and audit tracking.

Figure 2-2 on page 15 illustrates how the capabilities of IBM Security Access Manager for Enterprise Single Sign-On can be mapped to the IBM Security Blueprint. The diagram shows the functional components of the People and Identity Management solution pattern. The darker highlighted elements indicate the functional components that can be fulfilled or implemented by using IBM Security Access Manager for Enterprise Single Sign-On. This functional highlighting is also applicable for the infrastructure service components. For more information about the IBM Security Blueprint, see the IBM Redpaper™ publication *Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, REDP-4528.

In addition to the darker highlighted elements, Figure 2-2 on page 15 also shows medium highlighted elements. Although IBM Security Access Manager for Enterprise Single Sign-On can be used to address these components to some degree, the respective area of coverage is not considered a core function of the product and thus is considered to be limited.

[1] At this time, AccessAgent does not support an SMS or OTP token as second-factor authentication; it is supported for Web Assistant and Web Workplace only.

You might determine the desired function of a solution by using the People and Identity Management solution pattern. In this case, you can use the mapping shown in Figure 2-2 on page 15 as a quick reference of the functional security management aspects of IBM Security Access Manager for Enterprise Single Sign-On. This reference can help you determine which functions of a solution can be covered by selecting this product.

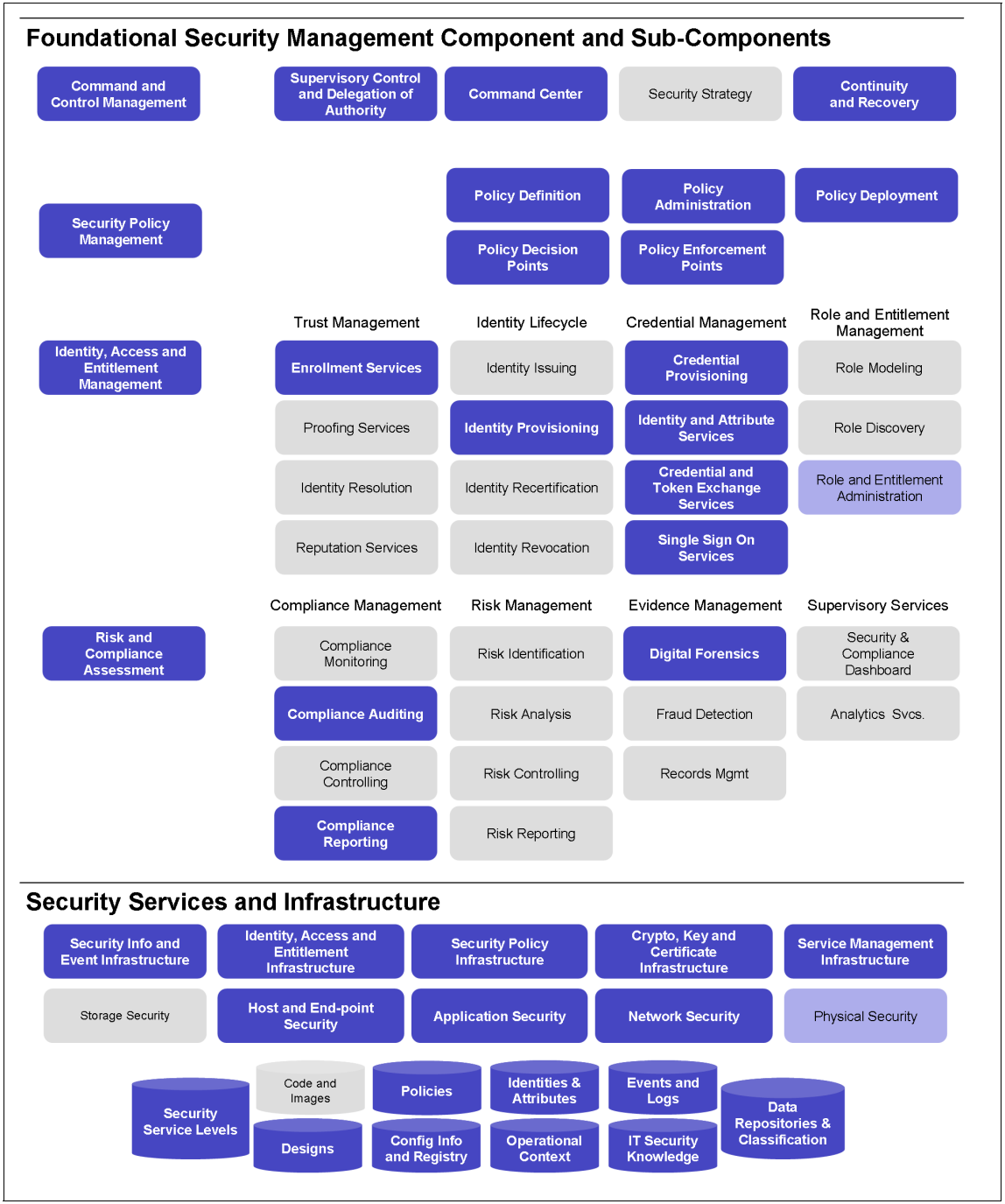

Figure 2-2 Mapping of the Access Manager for Enterprise Single Sign-On to the IBM Security Blueprint

IBM Security Access Manager for Enterprise Single Sign-On can help reduce costs, improve productivity, address compliance issues, and strengthen security with the following functions:

- Simplify deployment and management with a new virtual appliance configuration.
- Provide security for virtualized desktops and applications.
- Enhance strong authentication choices with support for hybrid RFID smart card, national ID cards, and finger biometrics.
- Facilitate regulatory compliance with fine-grained audit logs, centralized auditing, and reporting capabilities.
- Streamline user access with automated sign-on, sign-off, and a single password for all applications.
- Improve security and user productivity with comprehensive session management for kiosk or shared workstation environments.
- Reduce help desk costs with fewer password reset calls.
- Provide support for Microsoft Windows 7 64-bit platform and applications.

2.2 Logical component architecture

In this section, we introduce a logical component model of the IBM Security Access Manager for Enterprise Single Sign-On and describe every important component.

The logical component model illustrates the software components that are used to build a system. IBM Security Access Manager for Enterprise Single Sign-On consists of the following logical components:

- AccessAgent
- Microsoft Terminal Services, Citrix XenApp Server, or VMware vSphere virtual desktop AccessAgent
- IMS Server
- IMS Database
- AccessAdmin
- AccessStudio
- Provisioning Bridge

In the following sections, we describe every logical component in more detail. Figure 2-3 on page 17 depicts the overall logical component architecture.

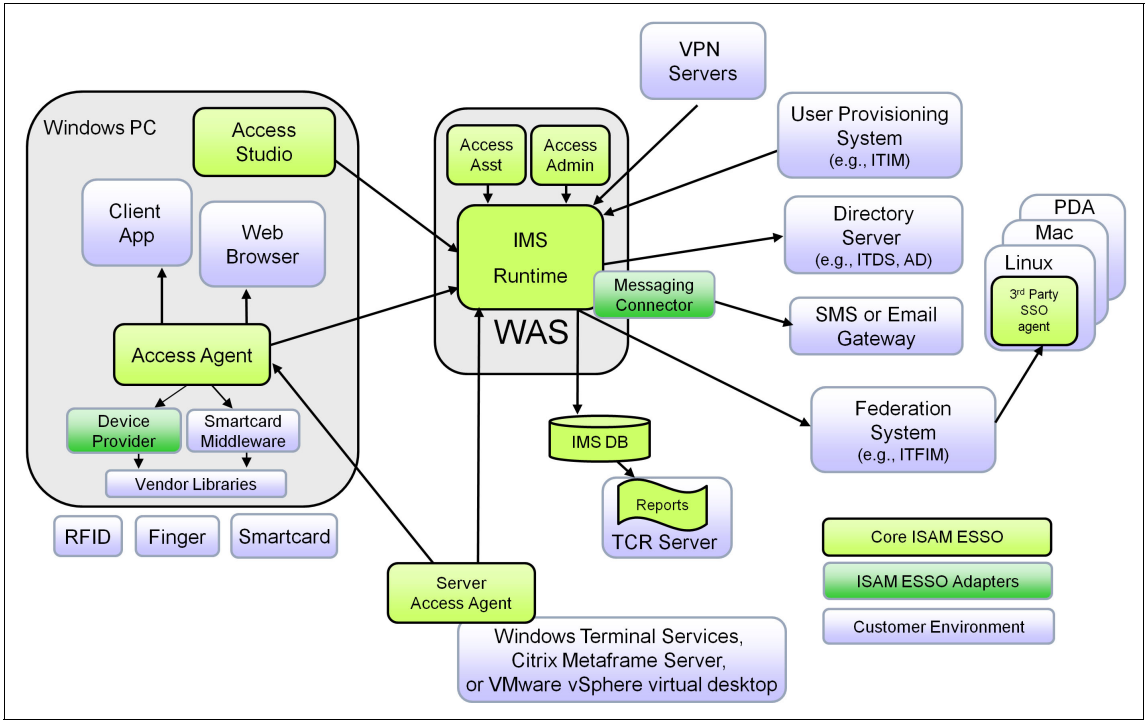

Figure 2-3 Logical component architecture

Acronyms: In the architecture that is shown in Figure 2-3, we use several product or component acronyms:

- IBM WebSphere® Application Server (WAS)
- IBM Tivoli® Identity Manager (ITIM)
- IBM Tivoli Directory Server (ITDS)
- IBM Tivoli Federated Identity Manager (ITFIM)
- Microsoft Active Directory (AD)
- IBM Tivoli Common Reporting (TCR)

2.2.1 AccessAgent

AccessAgent is the client software that is installed onto all Windows workstations and Terminal Servers or Citrix XenApp Servers and configured to connect to the designated IMS Server.

AccessAgent performs the following functions:

- AccessAgent can authenticate the user by configuring a combination of *authentication factors*. AccessAgent communicates with various devices (for example, an RFID reader, a smart card reader, and a biometric device) for each configuration. Optionally, AccessAgent can protect access to the Windows desktop by using these authentication factors and by replacing the native Microsoft GINA with the IBM Security Access Manager for Enterprise Single Sign-On GINA.
- AccessAgent can perform an automated user sign-on and sign-off to various applications. AccessAgent has an *Observer module* that is hooked into various applications. AccessAgent consults the appropriate AccessProfile (created by using AccessStudio) to perform the necessary logon and logoff and automation actions. AccessAgent stores the user credentials in a *Wallet*. By using the appropriate application credentials from this Wallet, AccessAgent can log users in to their applications.
- AccessAgent can manage multiple user sessions on the same workstation by using its *private desktop* feature.
- AccessAgent can track user application access activities and submit these audit events to the IBM Security Access Manager for Enterprise Single Sign-On IMS Server.
- AccessAgent can synchronize AccessProfiles, the credential Wallet of a user, and various policy settings with the IMS Server.
- AccessAgent provides a user interface for users to manage the application credentials stored in their credential Wallet, and their own IBM Security Access Manager for Enterprise Single Sign-On password and authentication factors.

Figure 2-4 on page 19 depicts the architecture of AccessAgent.

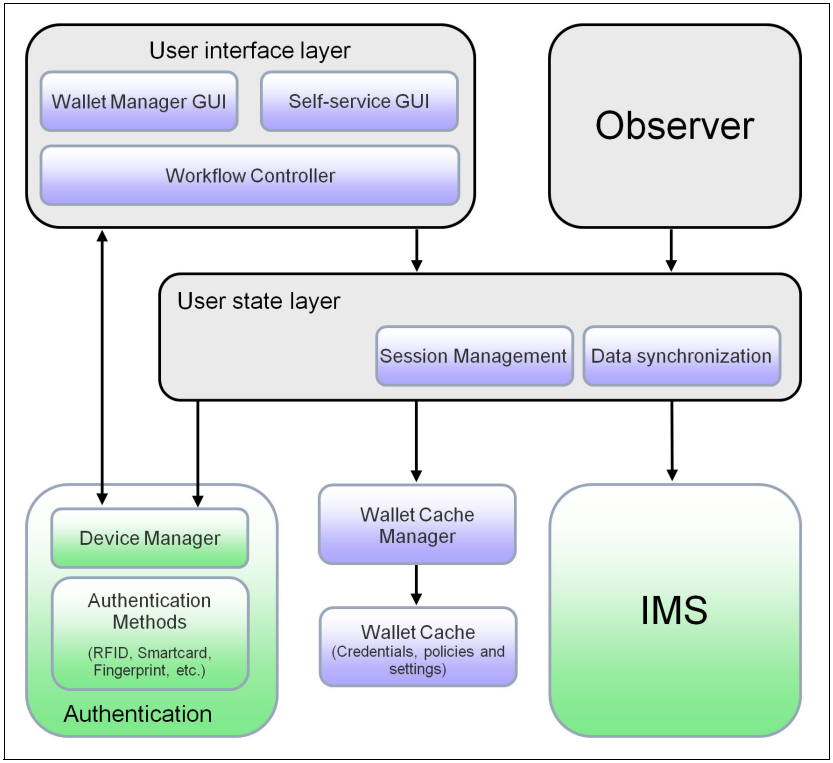

Figure 2-4 AccessAgent architecture

Let us take a closer look at the following AccessAgent function blocks:

- Authentication
- Data synchronization
- Wallet manager GUI
- Self-service GUI
- AccessAgent Observer module
- Workflow session management

Authentication

Authentication defines how the system validates users so they gain access to IBM Security Access Manager for Enterprise Single Sign-On, for example, by using a password, biometrics, or a token.

IBM Security Access Manager for Enterprise Single Sign-On supports the concept of a separation of the authentication of the user itself and the authentication against the Windows desktop. Let us describe what this separation means.

Introduction to desktop authentication and single sign-on

IBM Security Access Manager for Enterprise Single Sign-On maintains its own user repository and authenticates users by using various forms of authentication credentials stored in this repository. First, the user must be successfully authenticated by using IBM Security Access Manager for Enterprise Single Sign-On. Next, IBM Security Access Manager for Enterprise Single Sign-On is responsible for authenticating the user against the Windows desktop based on the Windows password of the user. The authentication against the Windows desktop can already be considered as a single sign-on from the IBM Security Access Manager for Enterprise Single Sign-On client to the Windows desktop of the user. The authentication method used for IBM Security Access Manager for Enterprise Single Sign-On can be a typical user name and password credential. However, it can also be an RFID token or smart card, for example, that is not supported by a standard Windows installation.

The authentication component, depicted in Figure 2-4 on page 19, consists of two layers:

- Authentication factors
- Authentication Device Manager

Authentication to IBM Security Access Manager for Enterprise Single Sign-On involves two steps:

1. The user provides credentials with the *authentication factors*.
2. The authenticator, for example a smart card or RFID reader, validates the user with the *Authentication Device Manager*.

Authentication factors

Authentication factors come in different forms and functions. Except for password and fingerprint, users can access systems and applications with a device that works like a key.

Let us first look at the basic factors:

- Password

  The *password* is used to secure access to a Wallet. The user specifies this password upon signing up with the IBM Security Access Manager for Enterprise Single Sign-On AccessAgent. Signing up with AccessAgent means registering the user with the IMS Server and creating a Wallet.

- Secret

  The user is asked to enter a *secret* when signing up for a Wallet. A secret is like a second password or a backup password. The secret must be something that the user does not forget, even if it is not used for a long time and it is not

likely to change. When the user signs up, the user selects a question from a list, and then provides the answer to that question. See Figure 2-5 for an example of providing a secret.

Figure 2-5 User enrollment secret question and answer

If a user forgets a password, the secret enables the user to set a new password. The user can also use the secret, along with an *authorization code*, to gain temporary access to the Wallet. An authorization code is generated by a help desk employee or an administrator. If self-service is enabled, users might have to specify a number of challenge-and-response questions during sign-up. Users can provide a subset of these challenge-and-response questions to perform password resets without using an authorization code.

Second authentication factors

Access to a user Wallet can be fortified with a *second authentication factor*. The combination of the password and a building badge or smart card, for example, strengthens the computer security of the user because both authentication factors must be presented to access the computer. Based on the security policy of the organization, the use of one of the following second authentication factors can be either mandatory or optional:

- RFID card

  The *RFID card* contains authentication information that can be retrieved through the use of a Radio Frequency Identification (RFID) reader. RFID works on the concept of proximity; the user taps the RFID card on the RFID

reader to gain access to credentials. The RFID reader is an additional piece of hardware that must be installed on every machine where the RFID card is used for authentication. An RFID card also allows for unified access. It can be used to access the computer, and for physical security (to access doors and elevators). Most companies today already use RFID cards for building and garage access. Now, the cards can also be used to authenticate users to the workstation.

- Active proximity badge

  The *active proximity badge* works almost identically as the regular RFID card, except that it contains a battery to enable the card to actively transmit security information to the associated reader. The active proximity badge can work over a greater range because of the active radio transmitter it contains. With the regular RFID card, the card must be close to the reader. With the active proximity badge, the distance can be specified. For example, the active proximity badge can be 2 meters (6.56 ft.) away from the reader, yet it is recognized. The reader automatically detects the presence of the user. For example, when the user leaves the workstation, AccessAgent locks the window, or logs off the user, depending on the policy setting.

- Fingerprint verification

  Fingerprint biometrics technology is generally recognized as the most cost-effective and convenient biometrics technology for enterprise-level deployment. The *fingerprint verification* system recognizes a fingerprint as an authentication factor. A fingerprint is matched to an earlier-registered fingerprint template stored in the IMS or cached at the user machine for authentication.

  Users use fingerprint authentication in place of (and not together with) a password to log on to AccessAgent.

- Smart card

  A *smart card* is a pocket-sized card that has an embedded microprocessor. Smart cards can do cryptographic operations, store, and process the digital credentials of the users securely. A smart card can be used as an authentication factor. Access Manager for Enterprise Single Sign-On provides certificate-based strong authentication when users access their Credential Wallet by using smart cards.

  For smart cards to work in Access Manager for Enterprise Single Sign-On, they must have cryptographic credentials. Smart cards must also have the corresponding certificate issued by either a corporate public key infrastructure (PKI) or trusted external PKI.

- Hybrid smart card

  Access Manager for Enterprise Single Sign-On supports the use of *hybrid smart cards* for user authentication in both personal and shared workstations.

Hybrid smart cards are made of an embedded PKI microprocessor with a contact interface and RFID chip with contactless interface. Users can log on and unlock the Windows desktop with a smart card without re-entering the smart card personal identification number (PIN) within a configurable grace period. The grace period is measured from the last two-factor authentication time.

Outside the grace period, the user logs on with the smart card and PIN through the contact interface. Within the grace period, the user can log on with the smart card only through the contactless interface across workstations. The logged on user can also unlock the Windows desktop with the smart card by using the contactless interface. The smart card PIN is not related to the Access Manager for Enterprise Single Sign-On password.

To use hybrid smart card authentication, the users must register the hybrid smart cards as a second authentication factor.

- Mobile ActiveCode

  The *ActiveCodes* are short-term authentication codes that are controlled by the IBM Security Access Manager for Enterprise Single Sign-On system. The *Mobile ActiveCode* is a randomly generated and event-based one-time password. The Mobile ActiveCode is generated on the IMS Server and delivered through a secure second channel, such as text services (SMS) on mobile phones or email. It is used to provide second-factor authentication in lieu of using key fob security devices. Mobile ActiveCode is used in an AccessAssistant and Web Workplace environment; it cannot be used to authenticate to AccessAgent.

By supporting building access badges and mobile devices for authentication, IBM Security Access Manager for Enterprise Single Sign-On is equipped to use *your existing security personal devices* as a second factor. For example, IBM Security Access Manager for Enterprise Single Sign-On enables the use of building access cards, such as the HID Prox, HID iClass, Mifare, and Indala cards, as second factors for logical access. This approach reduces the cost of acquisition, the cost of provisioning, and also the cost of support. It provides greater user convenience, relieving users from having to carry additional devices. User adoption is high and training costs are minimized because existing personal devices are used to secure access to corporate networks. Similarly, IBM Security Access Manager for Enterprise Single Sign-On can use existing fingerprint reader-enabled mobile computers and workstations for finger biometrics authentication.

IBM Security Access Manager for Enterprise Single Sign-On also enables secure remote access by combining two-factor authentication with leading Secure Sockets Layer (SSL) Virtual Private Network (VPN) platforms. With the solution, users can access web, desktop, and host-based applications through an

SSL-VPN connection and ensure two-factor authentication with one-time password (OTP) tokens or an OTP delivered to smartphones, PDAs, emails, or other mobile devices.

Regardless of the choice of authentication factors, administrators can centrally manage all authentication policies through the AccessAdmin interface. For Active Directory (AD) deployments, IBM Security Access Manager for Enterprise Single Sign-On can use the same password security policies as enforced by AD.

For non-AD deployments, administrators can enforce password policies on the IBM Security Access Manager for Enterprise Single Sign-On passwords set by users.

For information about policy settings for authenticators, see the *IBM Security Access Manager for Enterprise Single Sign-On Administrator Guide Version 8.2*, SC23-9951-03.

Authentication Device Manager

The *Authentication Device Manager* is used to integrate the authentication user interface with the main IBM Security Access Manager for Enterprise Single Sign-On AccessAgent. The Authentication Device Manager interfaces with the middleware and drivers of the device or reader to detect and read the second factor presented by the user.

Data synchronization

The *data synchronization* component synchronizes AccessProfiles, a user Identity Wallet, and various policy settings with the IMS Server and submits a user's application access audit events to the IMS Server. AccessAgent contacts the IMS Server on start-up, on each user login, and at periodic intervals, to synchronize data changes with the server. However, AccessAgent can cache data locally (on disk) based on a policy. AccessAgent can perform most of its functions even if it is disconnected from the IMS Server.

Wallet Manager GUI

The *Wallet Manager GUI* enables the user to manage the application credentials stored in the personal Identity Wallet. See Figure 2-6 on page 25 for an example of the Wallet Manager GUI.

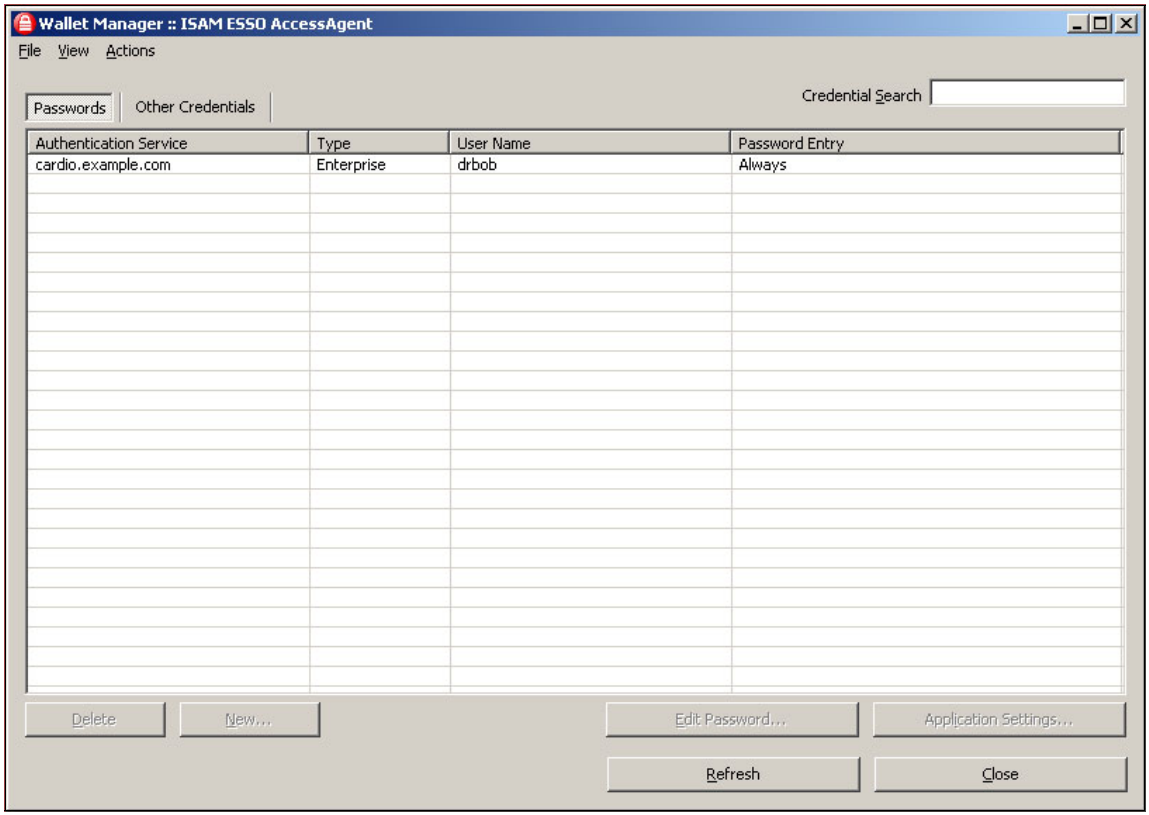

Figure 2-6 Wallet Manager GUI

Self-service GUI

A GINA extension is used to implement the *self-service user interface* for the user to manage the desktop password and authentication factors.

AccessAgent Observer module

AccessAgent Observer module is one of the core elements of IBM Security Access Manager for Enterprise Single Sign-On. The module is hooked into various applications, and consults the appropriate AccessProfile (created by using the AccessStudio application) to perform the necessary logon/logoff and automation actions. When an application presents a request for credentials, the Observer module is responsible for the appropriate action. The Observer module architecture is depicted in Figure 2-7 on page 26.

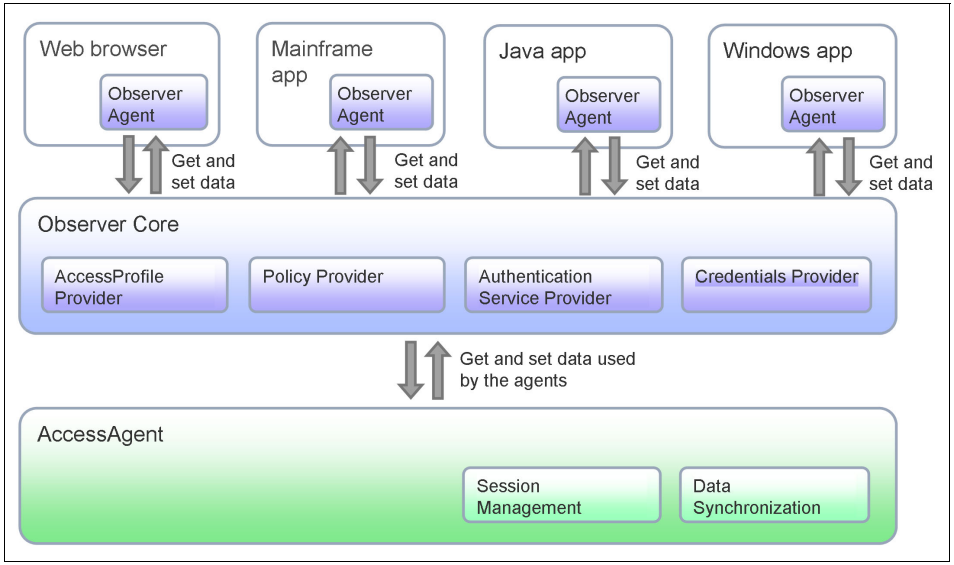

Figure 2-7 AccessAgent Observer module architecture

AccessAgent Observer module is composed of a core module and a number of agent instances that are hooked (through Windows APIs) into every launched Windows application, for example, the IBM Lotus Notes® application, Microsoft Outlook, and Microsoft Internet Explorer.

The Observer agent in each application can monitor its host application for various UI activities and to execute actions, such as to auto-fill a form field or auto-click a button as appropriate.

The behavior of AccessAgent *Observer agents* within each application is driven by a set of behavioral specifications called an *AccessProfile*. Each AccessProfile is an XML-structured file (based on a custom XML language) that provides a declarative set of preconditions. Figure 2-8 on page 27 is a graphical view of the resulting state machine of an *AccessProfile*.

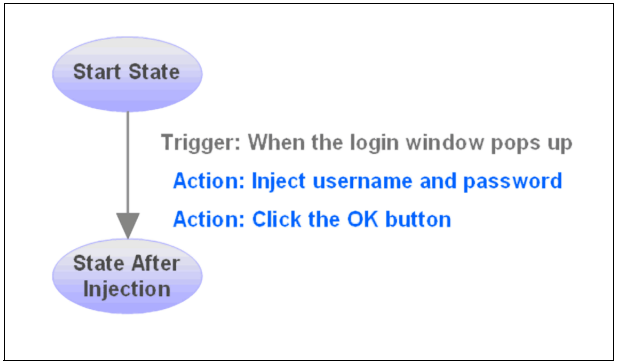

Figure 2-8 Workflow definition of a simple AccessProfile

Each *AccessProfile* entails a set of definitions for AccessAgent Observer agent module to watch for and execute:

- For Windows applications, the name of the executable
- A set of behavioral states, such as pre-logon or post-logon

  States represent specific situations where the state machine must look for certain triggers to occur (similar to a flowchart). A state can have multiple triggers. For example, in the `after_application_launched` state, you can look for the login window to appear or for a change-password window to appear. One trigger can have multiple actions. When a login window appears, you can inject user credentials and click OK. A profile writer can define as many states in a state machine as required.
- The following list shows the state definitions and with each state:
  - A set of workflow triggers (blue): *when*
  - Signatures that belong to a specific trigger: *where*
  - A set of workflow actions (red): *what*

The agent retrieves the required AccessProfiles, policies, authentication services, and user credentials from AccessAgent Observer core module. AccessAgent Observer core module, in turn, communicates with the rest of AccessAgent for data synchronization and workflow session management services.

A default set of AccessProfiles is provided with the IBM Security Access Manager for Enterprise Single Sign-On installer. However, each organization can create additional AccessProfiles for its own unique set of applications by using the AccessStudio tool.

Figure 2-9 depicts the Observer agent architecture. It consists of the following modules:

- Triggers
- Actions
- Application interface layer

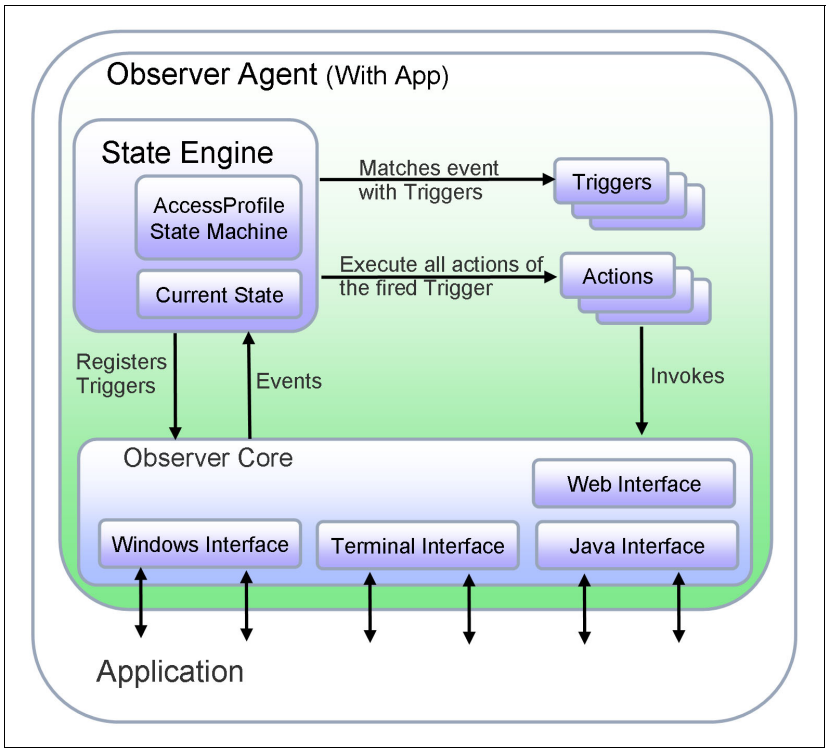

Figure 2-9 Observer agent architecture

Triggers

AccessAgent Observer agent module detects requests for credentials in various ways, depending on the application type (web, Windows, or mainframe/host). Triggers cause transitions between states in the state engine. A trigger defines *when* a condition is true.

The following list shows examples of triggers:

- `wnd_create_trigger` (Windows executable window is created.)
- `web_document_complete_trigger` (Web document completes loading.)
- `web_click_item_trigger` (HTML element is clicked.)
- `wnd_command_bn_click_trigger` (Windows executable button is clicked.)

Each trigger has a next-state defined. For example, when a login window is presented, the state machine can move to the `after_login_window_popped_up` state. Approximately 40 predefined triggers exist.

A trigger is specified by an equation *where* a specific trigger condition, such as a specific web URL or inside a specific application, must occur so that a condition is true.

Actions

An *action* can be performed in response to a trigger. That is, an action defines *what* must be done if a trigger becomes true. The following list shows examples of actions:

- `acc_data_inject_action` (Injection of credentials into defined fields)
- `acc_data_capture_action` (Capture of credentials from defined fields)
- `wnd_click_action` (Clicking a button in a Windows application)
- `acc_data_save_action` (Saving credentials to the Wallet)

Approximately 30 predefined actions exist.

Application interface layer

IBM Security Access Manager for Enterprise Single Sign-On includes Observer agents for different kinds of applications, and each application type has interface layers:

- Windows Interface
- Terminal Interface
- Web Interface
- Java Interface

In the following list, we look more closely at the various kinds of application interfaces and their related Observer agent modules:

- Windows Interface

  IBM Security Access Manager for Enterprise Single Sign-On responds to requests for user credentials from Windows applications. It works without any special configuration after you install it with the most widely used applications. In addition, you can configure it to work with any other individual application.

  All credential requests in Windows have specific attributes: application name, window name, and the control ID of the input field. IBM Security Access Manager for Enterprise Single Sign-On looks for the specific attributes of each application logon and password change dialog box and responds to these attributes. The attributes are stored in application profiles. For additional information, see the IBM Security Access Manager for Enterprise Single Sign-On AccessStudio.

The IBM Security Access Manager for Enterprise Single Sign-On AccessAgent Observer agent captures standard OS-level Windows messages and sends them to the IBM Security Access Manager for Enterprise Single Sign-On Observer core. When a specified application creates a dialog, IBM Security Access Manager for Enterprise Single Sign-On looks at the window title and other attributes as defined in the application profile. If IBM Security Access Manager for Enterprise Single Sign-On recognizes the required AccessProfile, it informs the Observer module to start the required actions.

IBM Security Access Manager for Enterprise Single Sign-On submits credentials to most Windows applications through secure, standard, OS-level Windows messages. Therefore, keyboard-sniffing utilities cannot intercept the credentials, and neither can users confuse the response by moving the mouse or clicking the keyboard.

- Terminal Interface

  IBM Security Access Manager for Enterprise Single Sign-On responds to requests for user credentials from mainframe and host applications. It works without modification with the most popular mainframe and host emulators. In addition, you can configure it to work with others.

  All requests for credentials in mainframe and host applications have specific attributes: window title and various blocks of text (at specific coordinates for mainframe applications), user name and password field text, and so on. IBM Security Access Manager for Enterprise Single Sign-On looks for the specific attributes of each application's logon and password-change windows and responds accordingly. The attributes are stored in the application profiles, as well. For additional information, see the IBM Security Access Manager for Enterprise Single Sign-On AccessStudio.

  The IBM Security Access Manager for Enterprise Single Sign-On Mainframe application Observer agent monitors emulators, looking for the defined matches. When a new panel is presented, IBM Security Access Manager for Enterprise Single Sign-On reviews the text for matching fields. If all attributes match, the Observer module submits the user credentials.

  IBM Security Access Manager for Enterprise Single Sign-On submits credentials to most emulators through high-level language application programming interface (HLLAPI). Therefore, keyboard-sniffing utilities cannot intercept the credentials, and neither can users confuse the response by moving the mouse or clicking the keyboard.

- Web Interface

  IBM Security Access Manager for Enterprise Single Sign-On responds to requests for user credentials from web applications, whether in a form or through a pop-up dialog. Unlike most single sign-on products, IBM Security

Access Manager for Enterprise Single Sign-On supports access to all web applications, not just intranet applications.

All credential requests in web applications are either in *forms* or in *dialog boxes*. The IBM Security Access Manager for Enterprise Single Sign-On Web Browser Observer agent responds to the specific events of a web dialog box opening or of a web page rendering.

Because the Observer does not use keystrokes for web browsers, users cannot confuse the response by selecting and working in another application.

- Java Interface

  IBM Security Access Manager for Enterprise Single Sign-On responds to login and password change requests for virtually all Java applications and applets built on the Sun Java Runtime Engine 1.4.1 or higher. New Java applications or applets can be supported by using the IBM Security Access Manager for Enterprise Single Sign-On AccessStudio.

AccessAgent Plug-In

AccessAgent Plug-In is a block of VBScript or JavaScript code that performs some custom action needed as part of a workflow trigger or workflow action inside an AccessProfile. This block of code can make calls into the Windows OS, and into an AccessAgent Plug-In API, by using the Windows and IBM Security Access Manager for Enterprise Single Sign-On privileges of the user.

Administrators typically use this extension facility to implement customized authentication, access control, or workflow automation for a specific application:

- Retrieve an application credential (for the current application) from the user's Wallet.
- Retrieve a user policy setting from a user's Wallet.
- Look up a user's group membership or attribute from the user directory.
- Read or store data from a central fileshare.
- Look up the time from the host system clock.
- Perform an additional checksum or check the installation path on a target application before the single sign-on.
- Call an external application or process.
- Make an HTTPS call to a third-party service.

Session management

IBM Security Access Manager for Enterprise Single Sign-On supports two main usage configurations: *personal workstations* and *shared workstations*. The personal workstation configuration is typically used in organizations where users

are assigned their own workstations. The shared workstation configuration, for example, is in healthcare organizations where doctors and nurses share workstations that are deployed throughout the hospital. IBM Security Access Manager for Enterprise Single Sign-On supports *fast user switching* through any of the following schemes:

- *Shared desktop*
- *Private desktop*
- *Roaming desktop*

We describe the supported modes for shared workstations in more depth:

- Fast user switching through shared desktop

  Shared desktops allow multiple users to use one generic Windows desktop in a workstation. Because each user does not have to log on to Windows, switching users is quicker. However, after switching from user A to user B, the application contexts for user A are lost. If user A returns later and switches the workstation back to user A's account, the user must restart the applications. For the scheme, AccessProfiles must be created to automatically log off enterprise applications when user switching occurs.

- Fast user switching through private desktop

  Private desktops allow multiple users to have their own Windows desktops in a workstation. The scheme uses the *local user session management* feature of AccessAgent on Windows XP and the Fast User Switching feature for Windows Vista and Windows 7. These features allow users to retain the existing user's desktop session when switching users. When user A returns to the workstation to unlock it, AccessAgent switches to user A's earlier desktop session, allowing user A to resume the previously incomplete or interrupted work.

- Fast user switching through roaming desktop

  Roaming desktops provide users with Windows virtual desktops to *roam* to their points of access, from workstation to workstation. With roaming sessions, a user can disconnect from the current virtual desktop or application session at a client, log on to another client, and continue the desktop or application session at a new client. The scheme requires the use of either a Microsoft Windows Server Terminal Services session or Citrix XenApp Presentation Server session.

If AccessAgent is configured for shared workstation operation, the workflow session management module is responsible for the desktop switching between the different users.

2.2.2 AccessAgent in server mode

AccessAgent includes a *server mode* that is automatically enabled when deployed on a Microsoft Terminal Services or Citrix XenApp Server. A separate instance of the server-side AccessAgent is started for each terminal session. When a new terminal session is started, the server-side AccessAgent looks for the client-side AccessAgent across a virtual channel established between the terminal server and its Remote Desktop Protocol (RDP) client. Then, the server-side AccessAgent instance retrieves the required data (profiles, credentials, and policies) from the client-side AccessAgent to perform various single sign-on operations within the terminal session. If the user captures a new credential at either the client or server session, this change is immediately visible at the other end.

To enable the virtual channel communication feature for a Citrix XenApp server environment, a service engagement is necessary to provide the applicable software. AccessAgent in server mode can also function in a stand-alone manner without relying on a virtual channel to a client-side AccessAgent. In this scenario, AccessAgent in the terminal session synchronizes the user's single sign-on data directly with the IMS Server. No service engagement is required in this case.

Server mode means "server": Server mode is applicable only on a Windows Server OS and not on a Windows client OS, such as Windows XP or Windows 7.

2.2.3 IMS Server

The Integrated Management System (IMS) Server provides a central point of administration and control. It enables centralized management of *user identities*, *AccessProfiles*, and *authentication policies*. It also provides *loss management* of authentication tokens and *audit management*. The IMS Server interfaces with other applications through *IMS Connectors* and *IMS Provisioning Bridges*. The IMS Server can be configured with *AccessAdmin*. Lower-level configuration settings for the IMS Server can be configured with the *IMS Configuration Utility*, which is accessible by administrators. The IMS Server exposes an internal SOAP API that is used by *AccessAdmin*, *AccessStudio*, and *AccessAgent*.

Figure 2-10 on page 34 shows the IMS Server architecture.

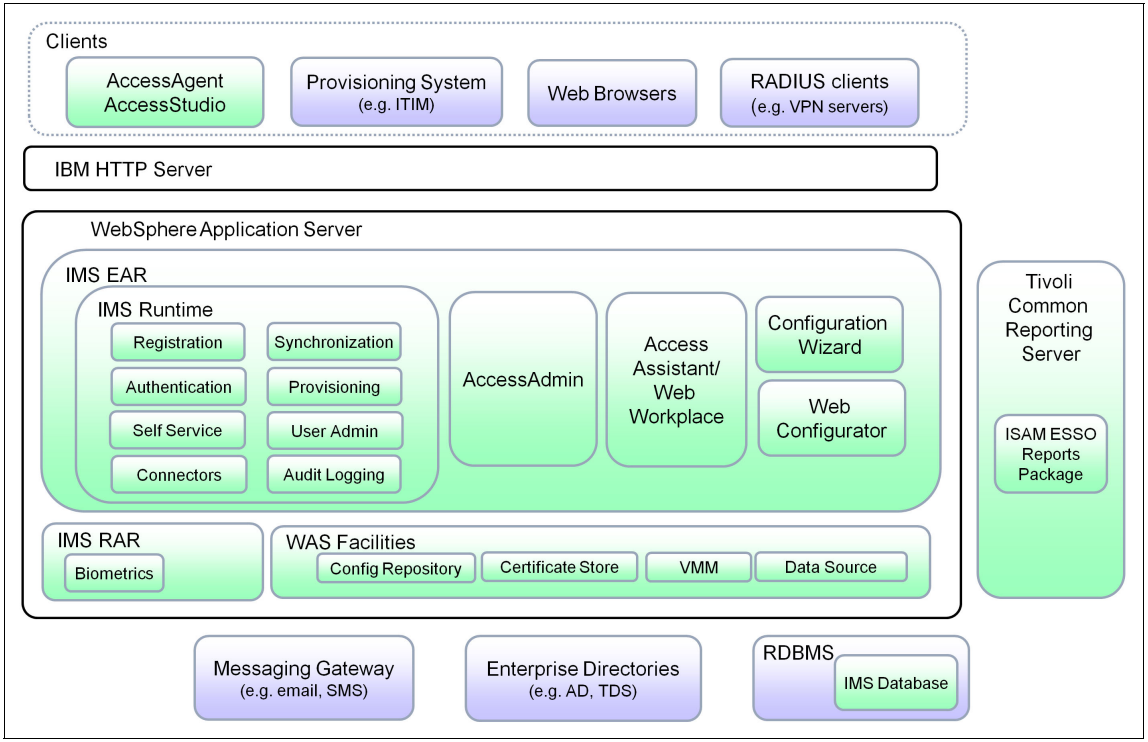

Figure 2-10 IMS Server architecture

The IMS Server application runs on the IBM WebSphere Application Server platform; it is bundled into web application archive (WAR) files and further packaged into an enterprise archive (EAR). This EAR file must be deployed by using the IBM Security Access Manager for Enterprise Single Sign-On installer. The WAR files contain the following main modules of the IMS Server:

- IMS Runtime

  This module implements the core IMS logic and exposes SOAP APIs for AccessAgent, AccessAssistant/Web Workplace, and third-party provisioning systems. This module can be categorized into the following subcomponents:

  – Identity management

    The IMS Server provides basic identity management functions, such as user enrollment and password management for users and administrators. Supported by a self-service module, users can manage their own credentials, such as resetting their password.

– Authentication

  The IMS Server provides a one-time password mechanism called ActiveCode. This ActiveCode is a strong authentication mechanism to authenticate users online or when their desktop has no connection to the IMS Server. To allow virtual private network (VPN) servers to authenticate with a one-time password, the IMS Server also provides a RADIUS interface.

– Auditing

  The auditing framework captures identity information and events in the database to allow administrators to generate reports for identity auditing:

  - List of application accounts for a user
  - Policy changes performed on a user by an administrator or help desk
  - Successful and failed application logons and logoffs
  - Summary table of the number of times each user logs on to each application within a certain period

  To analyze the audit log, administrators can use four standard reports that are provided with the Tivoli Common Reporting engine. Or, they can generate identity auditing reports by using an SQL query tool (for example, Microsoft Excel, Microsoft SQL Query Analyzer, and Crystal Reports).

– Other services

  The IMS Server synchronizes AccessProfiles, a user's Identity Wallet, and various policy settings with AccessAgents. It receives the user's application access audit events from AccessAgent.

- AccessAdmin

  AccessAdmin is the administrative UI for the IMS Server, which is described in more detail in 2.2.5, "AccessAdmin" on page 37.

- AccessAssistant

  The AccessAssistant is the web-based interface used to provide password self-help and password reset services to users. The AccessAssistant module is packaged into a WAR that is deployed with the IMS Server application. AccessAssistant uses a SOAP API to authenticate and execute its self-service and Wallet management services.

- Web Workplace

  The Web Workplace is the web-based interface that provides users with the ability to log on to enterprise web applications by clicking links, without needing to remember the passwords for individual applications. The Web Workplace is described in more detail in "Web Workplace" on page 42.

- IMS Configuration Utility

  This utility is a web-based interface for configuring the different IMS Server settings.

- IMS Configuration Wizard

  This wizard hosts the first-time start-up configuration pages for the IMS Server (where database and certificate attributes are configured).

The IMS Server uses the certificate support infrastructure of its WebSphere Application Server host. It uses the same certificate authority and Java truststore for secure communication. Any designated certificate authority (CA) can sign IBM Security Access Manager for Enterprise Single Sign-On IMS certificates, which provides flexibility.

The IMS Server also uses the WebSphere Application Server configuration repository to store its configuration data (for example, the `ims.xml` file). Therefore, you can now rely on WebSphere Application Server Network Deployment to automatically replicate configuration changes to all IMS Server application instances that run in a WebSphere Application Server cell.

The IMS Server relies on an external relational database (IBM DB2®, SQL Server, or Oracle) to store its system data and user data. The IMS Server also uses the WebSphere Application Server data source (and connection pooling) facilities to connect to the IMS database. You have additional flexibility to tweak the IMS connection to its database, if required, through the WebSphere Application Server administrative console. Also, you can use the WebSphere Application Server Performance Monitoring Infrastructure (PMI) infrastructure to view performance and monitoring counters for IMS data sources.

The IMS Server relies on the Virtual Member Manager (VMM) facility in WebSphere Application Server to perform the lookup and verification of users. Therefore, you have the flexibility to use the VMM broad support for various directories.

2.2.4 IMS database

The IMS Server relies on an external relational database to store its system data and user data. It also stores all its audit logs in the same or a separate database instance. The IMS Server application communicates with the database by using Java Database Connectivity (JDBC).

2.2.5 AccessAdmin

The AccessAdmin component is the web-based management console used by administrators and help desk employees to manage users and policies on an IMS Server (Figure 2-11). Different access rights are granted to the administrator and help desk roles. Certain configurations (for example, system policies) can be viewed but not modified by the help desk staff.

The AccessAdmin GUI provides the following functions:

- User search and administration (to modify user policies, issue authorization code, unlock a locked Wallet, revoke a user, and so on)
- Create and maintain policy templates (can be created and maintained by an administrator only, but help desk staff can view and apply)
- Set system and application policies (can be modified by an administrator only, but help desk staff can view)
- Access logs and status information

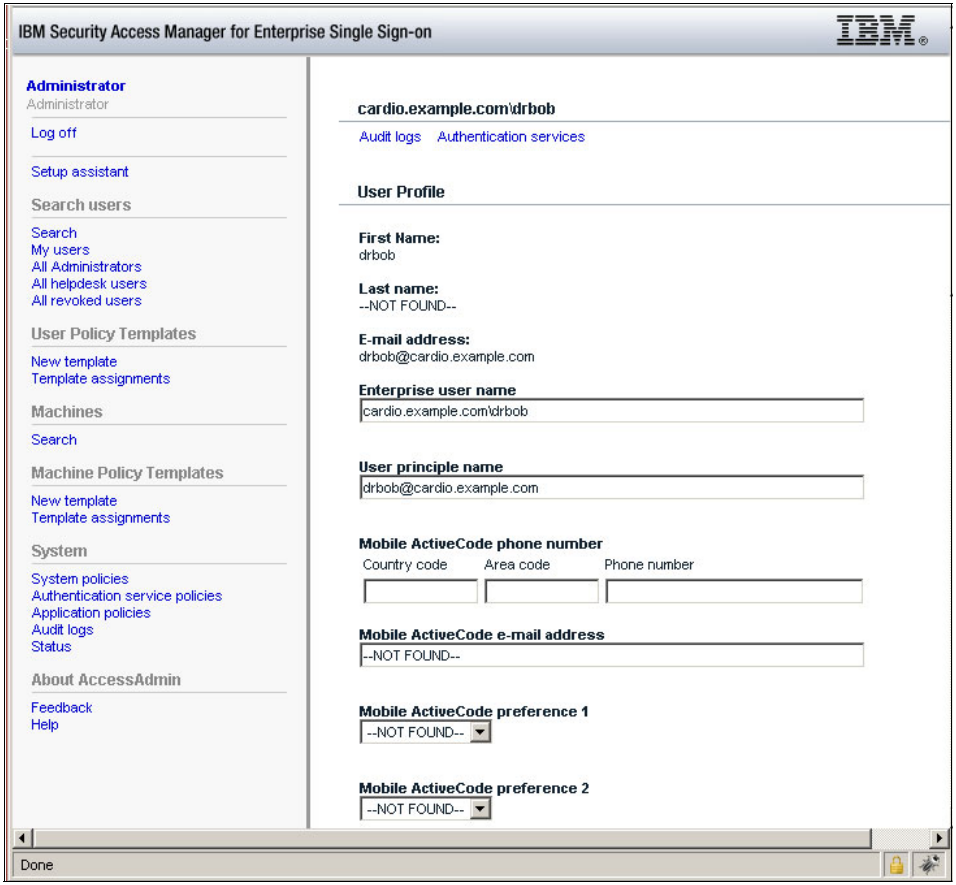

Figure 2-11 Web-based administration by using AccessAdmin

Authorization and access management

Role-based access control is used to protect access to operations in the AccessAdmin, Web Workplace, and AccessAssistant applications. Users are classified into three roles:

- User
- Administrator
- Help desk

A new user is granted the user role by default, when the user first registers to the IMS Server through AccessAgent or AccessAssistant.

Only users with the administrator role have full access to the AccessAdmin and AccessStudio applications. For example, only administrators can upload new or modified AccessProfiles at the IMS Server and can modify the system through

AccessAdmin. The first administrator user is provisioned during the IMS Server configuration, by using the IMS Configuration Utility. Then, an administrator user can assign any user the administrator or help desk role through the AccessAdmin interface.

Help desk members are responsible for user administration tasks, such as managing user policies, revoking a second-factor token, such as a USB token or smart card, and issuing an authentication token for password reset or second-factor registration.

2.2.6 AccessStudio

The AccessStudio application is used by administrators to create AccessProfiles required to support sign-on/sign-off and custom workflow automation. This component can run on Windows only (Figure 2-12 on page 40).

The AccessStudio application provides these functions:

- A wizard mode is for administrators to easily generate AccessProfiles for most applications, by walking through the set of application windows and by mapping selected fields/controls used for logon, logoff, and other application behaviors.
- An advanced mode is for administrators to create AccessProfiles for complex applications or where complex workflow automation is required.
- A test mode is for administrators to test a generated AccessProfile against the target application.
- When the AccessProfile is finished, it can be uploaded to the IMS Server with AccessStudio.

The AccessStudio must be installed on an existing AccessAgent installation. The user must have an administrator role and must have an active AccessAgent session before downloading from or uploading to the IMS Server is possible.

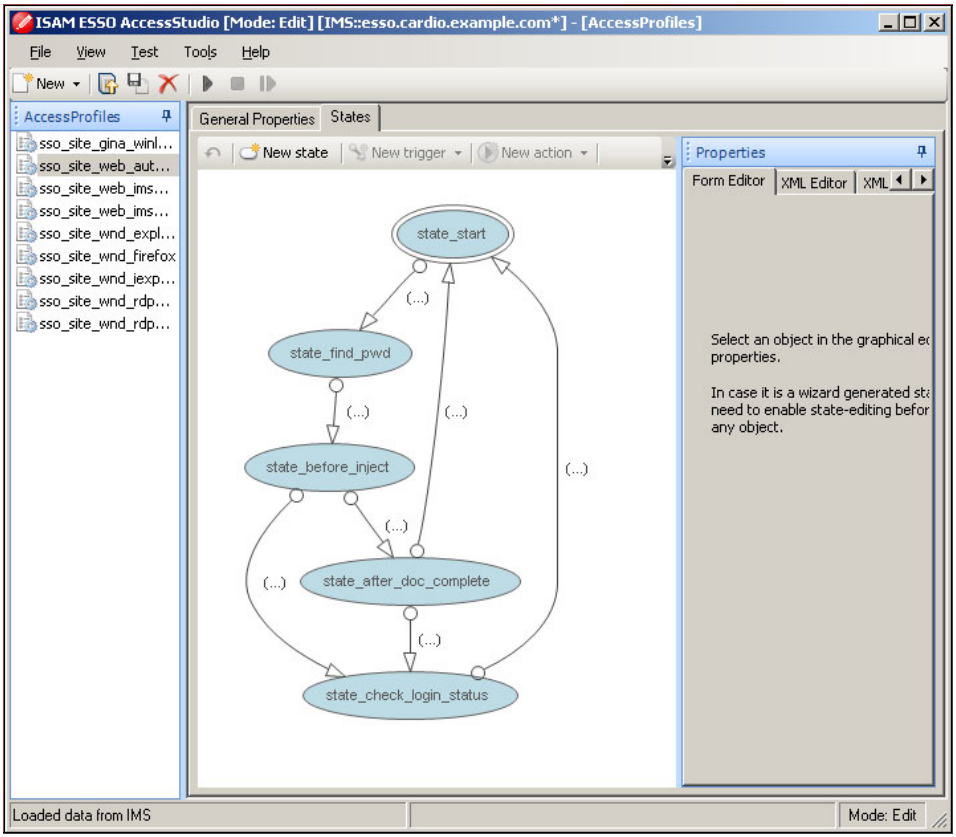

Figure 2-12 AccessStudio application

2.2.7 Provisioning API

The IMS Server provides a *provisioning API* that can automate the user credential distribution process so that identity management solutions, such as Tivoli Identity Manager, can provision and remove user involvement in the credential provisioning and management process. It enables an administrator to automatically provision IBM Security Access Manager for Enterprise Single Sign-On with a user ID and password by using an external provisioning system. An administrator is able to *add*, *modify*, and *delete* IDs and passwords for particular applications within the provisioning system and have the changes reflected in IBM Security Access Manager for Enterprise Single Sign-On.

From the provisioning system, all user names and passwords in IBM Security Access Manager for Enterprise Single Sign-On can also be deleted so that a user's access to all protected applications is revoked.

In most organizations, users must know, remember, and enter their application credentials. This responsibility is a particular burden on the first day a user begins work or takes on a new set of responsibilities and permissions. But when an organization uses the IBM Security Access Manager for Enterprise Single Sign-On provisioning API, application credential provisioning and deprovisioning between Tivoli Identity Manager and IBM Security Access Manager for Enterprise Single Sign-On are automated. So, organizations no longer must physically distribute credentials to users who must enter them manually into IBM Security Access Manager for Enterprise Single Sign-On.

Instead, administrators directly create, edit, and delete user credentials through Tivoli Identity Manager. Users can enjoy single sign-on from day one and are no longer responsible for tracking their own application credentials, while helping to maximize security. When users no longer need access to systems, the integration between the applications enables Tivoli Identity Manager to remove or revoke the users' systems and application access and also delete their credentials automatically from the IBM Security Access Manager for Enterprise Single Sign-On data store. Controlling the appropriate level of access helps maximize security and assists with compliance initiatives by demonstrating enforcement of internal controls to auditors.

Furthermore, the IBM Security Access Manager for Enterprise Single Sign-On provisioning bridge provides a high level of administrative control. For example, when application passwords are reset in Tivoli Identity Manager, IBM Security Access Manager for Enterprise Single Sign-On is simultaneously updated so that it always has the correct password. Additionally, it extends audit and reporting capabilities to include information about applications and the use of applications that are configured in IBM Security Access Manager for Enterprise Single Sign-On but that fall outside Tivoli Identity Manager.

2.3 Additional components

IBM Security Access Manager for Enterprise Single Sign-On also includes the following additional modules:

- AccessAssistant

  The AccessAssistant is a web-based interface that enables users to manage their Identity Wallet. They can reset their IBM Security Access Manager for Enterprise Single Sign-On password and change the reset questions and answers. They can view, add, edit, or delete user names and passwords inside their Wallet.

- Web Workplace

  The Web Workplace component provides a web-based interface that enables the user to log on to enterprise web applications by clicking links, without needing to remember the passwords for individual applications. Web Workplace is often deployed as part of an organizational SSL VPN portal. Users can (by using a web browser) use SSO in enterprise web applications with the help of Web Workplace technology. Users also can connect into a Citrix XenApp or Microsoft Terminal Services session and use SSO with the AccessAgent instance that runs within the session.

- Web APIs

  The Web APIs allow a developer to write a custom enterprise single sign-on Agent for various platforms, such as Apple Mac OS, Linux, and smartphone, that can interoperate with the IMS Server. In particular, with these Agents, you can get, set, and delete user credentials on the IMS Server from any platform. For more information, see the *IBM Security Access Manager for Enterprise Single Sign-On Web API Guide for Credential Management Version 8.2*, SC14-7646.

2.4 Physical component architecture

In this section, we describe the physical components that are assembled for IBM Security Access Manager for Enterprise Single Sign-On. Figure 2-13 on page 43 shows a simple, base deployment architecture. For a detailed description of different server deployment scenarios, see 3.3, “Deployment strategies” on page 75.

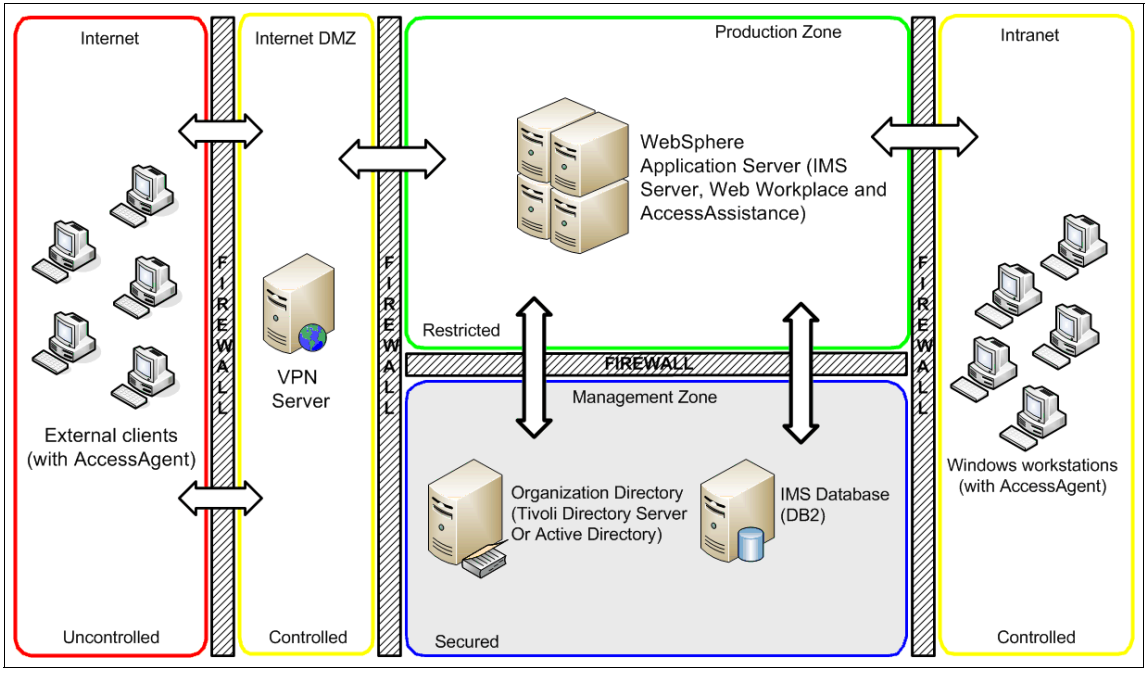

Figure 2-13 Physical base deployment architecture

2.4.1 IMS Server

The IMS Server serves as the central repository and management point for all system and user data consumed by AccessAgents. The IMS Server performs the following functions:

- Serves as a central repository and distribution point for AccessProfiles and other system data.
- Serves as a central repository for all user data, including the credential Wallet and various authentication and access policies.
- Provides a SOAP API for AccessAgents, AccessAssistant, and Web Workplace servers to authenticate users, and to retrieve and synchronize system and user data.
- Provides a SOAP API for AccessStudio to upload new or updated AccessProfiles for distribution to AccessAgents.

- Provides a SOAP API for Tivoli Identity Manager to provision application credentials into user Wallets and users into the IMS Server.
- Provides SOAP and RADIUS APIs for third-party software, such as VPN, to authenticate users through one-time passwords.
- Provides a web-based interface for administrators to manage users, machines, and system policies, and to query audit logs. The web-based interface is named AccessAdmin.

The IMS Server consists of a group of web-based applications developed in Java and run on top of WebSphere Application Server. Administration of the WebSphere Application Server is not necessary during IMS Server operation.

2.4.2 IMS database

The IMS Server stores all its data within a relational database. The IMS database contains these classes of data:

- System data

  The class of system data includes AccessProfiles, system policies, user and machine policy templates, and other system configuration data.

- User data

  The class of user data includes application credentials and user policies.

- Machine data

  The class of machine data includes any machine policies and information about deployed machines.

- Audit logs

  Every user and administration activity is stored in the database.

2.4.3 Organization directories

An enterprise directory can be used to manage user accounts, including those accounts maintained by the IMS Server.

The integration of organization directories

An *organization directory* is an entity that validates user credentials for IBM Security Access Manager for Enterprise Single Sign-On users. It can be used for validating users during signup and also during logon, if the password is set up to synchronize with the enterprise directory password. It can be a directory of user accounts that define IBM Security Access Manager for Enterprise Single

Sign-On users. An example for an enterprise directory can be an AD forest, as depicted in Figure 2-14.

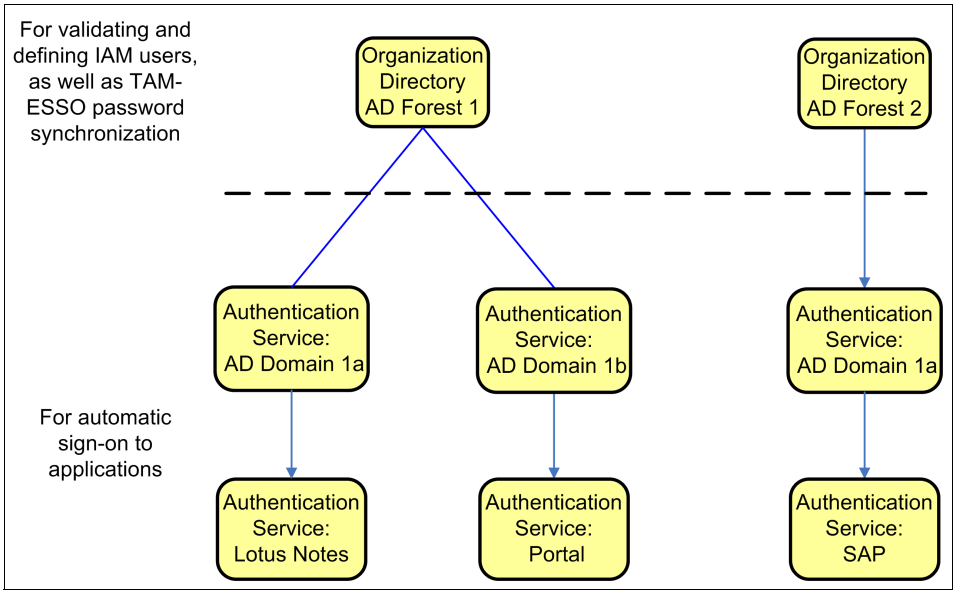

Figure 2-14 Organization directory integration

An organization directory can contain several authentication services, or none at all. An AD forest with multiple domains can be an enterprise directory that contains multiple authentication services, with each authentication service representing one domain. This definition, coupled with the password synchronization feature, allows enterprise directory passwords to be used for both logon to the Wallet and automatic sign-on to applications.

Applications, such as Lotus Notes®, a portal, or SAP applications can be tied to one, more, or all AD authentication services in a deployment.

Use of existing user registries

IBM Security Access Manager for Enterprise Single Sign-On uses existing user registries (for example, Microsoft AD or IBM Tivoli Directory Server) to identify and validate a user when the user registers or signs up. After this step, it creates an account for this user in its own user repository (stored on the IMS database). Thereafter, only this database is consulted during run time when the user accesses the IBM Security Access Manager for Enterprise Single Sign-On functions. Additionally, user accounts can be provisioned into IBM Security Access Manager for Enterprise Single Sign-On by using user provisioning products, such as Tivoli Identity Manager.

For deployments where the IMS Server is configured to use Microsoft AD as its user repository, IBM Security Access Manager for Enterprise Single Sign-On can be configured to perform password synchronization with AD. In this configuration, users can always log on to AccessAgent with their latest AD credentials. If this AD password is reset out-of-band, AccessAgent and the IMS Server verify the new AD password against the AD server. And, they resynchronize the IBM Security Access Manager for Enterprise Single Sign-On password to this new value.

Additionally, for AD deployments, the IMS Server can additionally look up the directory for attributes of Windows workstations joined to the domain, and use these attributes to select a machine group policy template to apply onto the machine.

2.4.4 AccessAgent

AccessAgent gets deployed on user and administrator workstations either manually or by using software distribution mechanisms. It is suggested that the AccessAgent installer is preconfigured by the IT organization so that the installation involves no or minimal user intervention.

AccessAgent and GINA/Credential Provider chaining

For AccessAgents installed with the GINA (for Windows XP) or Credential Provider (for Windows Vista and Windows 7) option enabled, a user logs on to the AccessAgent GINA/Credential Provider first, with the required authentication factors. Then, AccessAgent auto-logs on the user to Windows by using the Windows account of the user. The Windows GINA/Credential Provider is not replaced and is still available, as needed.

For AccessAgents installed without the GINA/Credential Provider replacement, seamless logon is possible (for AD deployments) by using the EnNetworkProvider feature. For non-AD deployments, the user must separately log on to AccessAgent after logging on to Windows.

Availability constraints

If AccessAgent has network connection to the IMS Server, it authenticates a user against the IMS Server by passing along the authentication credentials over HTTPS to the IMS Server. However, if AccessAgent is offline to the IMS Server, it then authenticates the user's presented credentials against cached authentication data stored on the disk.

2.5 IBM Security Access Manager for Enterprise Single Sign-On integration

In this section, we describe the integration of IBM Security Access Manager for Enterprise Single Sign-On with other products. Figure 2-15 illustrates the basic integration with other products along with communication protocols.

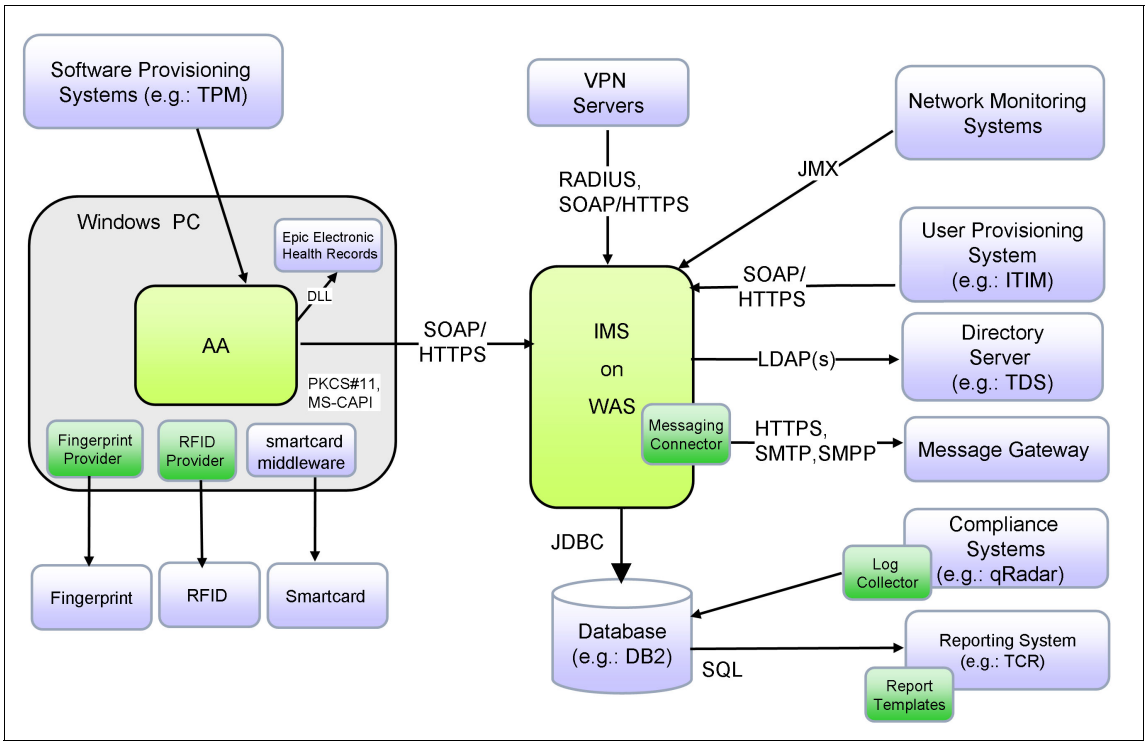

Figure 2-15 IBM Security Access Manager for Enterprise Single Sign-On integration

2.5.1 User provisioning products

Products, such as Tivoli Identity Manager, can use the IBM Security Access Manager for Enterprise Single Sign-On provisioning APIs to provision users and their accounts for various applications within IBM Security Access Manager for Enterprise Single Sign-On. These products typically provide a self-care interface for users to manage operations on their accounts, such as password reset and password change. You can integrate IBM Security Access Manager for Enterprise Single Sign-On with these products for self-service functions.

2.5.2 Compliance products

IBM Security Access Manager for Enterprise Single Sign-On audit logs can be used by compliance products, such as the IBM Security QRadar family, to generate cross-product compliance reports. With compliance products, you can archive security intelligence to make it an intelligent, integrated, and automated solution.

2.5.3 Software provisioning products

IBM Security Access Manager for Enterprise Single Sign-On AccessAgent can be provisioned to all Microsoft Windows workstations by using products, such as IBM Tivoli Provisioning Manager.

2.5.4 Web single sign-on

IBM Security Access Manager for Enterprise Single Sign-On can provide single sign-on to back-end single sign-on systems, such as Tivoli Access Manager for e-business with its WebSEAL component. Tivoli Access Manager for e-business with its WebSEAL component, in turn, can provide single sign-on and centralized access control for web applications.

2.5.5 User repositories

IBM Security Access Manager for Enterprise Single Sign-On can integrate with various user repositories (for example, Microsoft AD or IBM Tivoli Directory Server) to determine the validity of a user when the user initially registers to IBM Security Access Manager for Enterprise Single Sign-On.

2.5.6 Database servers

IBM Security Access Manager for Enterprise Single Sign-On can integrate with various database vendors (IBM DB2, Oracle, and Microsoft SQL) to store its user, policy, and audit information in a database.

2.5.7 Reporting tools

Reports on IBM Security Access Manager for Enterprise Single Sign-On audit log and user data can be generated through any third-party SQL Reporting tool or framework. IBM Security Access Manager for Enterprise Single Sign-On also provides standard reports by using its Tivoli Common Reporting module.

2.5.8 Monitoring products

The IBM Security Access Manager for Enterprise Single Sign-On IMS Server provides Java Management Extensions (JMX) APIs that can be used for monitoring the deployment, such as IBM Tivoli Monitoring. For more information about JMX APIs, see the *IBM Security Access Manager for Enterprise Single Sign-On Configuration Guide Version 8.2*, GC23-9951-01. Because the IMS Server is a WebSphere Application Server application, you can also monitor it like any other WebSphere Application Server application.

2.5.9 Third-party readers

The IBM Security Access Manager for Enterprise Single Sign-On AccessAgent can work with any RFID reader as long as a Provider module based on the RFID Reader Serial Peripheral Interface (SPI) is developed and deployed on the host machine.

IBM Security Access Manager for Enterprise Single Sign-On AccessAgent works with fingerprint readers from all major manufacturers (including readers already embedded in many notebook products) when the BIO-key Biometric Service Provider is deployed.

2.5.10 Epic Electronic Health Records

IBM Security Access Manager for Enterprise Single Sign-On integrates with Epic through an Epic-defined interface for sign-on automation to all applications that support patient care. When the user starts Epic, it executes the callback function implemented by IBM Security Access Manager for Enterprise Single Sign-On in the form of a DLL. The callback function retrieves the user credentials from IBM Security Access Manager for Enterprise Single Sign-On and then passes them to Epic. Epic authenticates the user and the user is logged in.

2.6 Conclusion

We conclude the architecture and component design for IBM Security Access Manager for Enterprise Single Sign-On. We looked closely at the internal data flow between the different logical components. By using physical component diagrams, we provided descriptions of where to deploy the physical components. We also discussed the integration of IBM Security Access Manager for Enterprise Single Sign-On with other products.

3

Solution design and management

In this chapter, we highlight important aspects involved in the design of an IBM Security Access Manager for Enterprise Single Sign-On solution. When determining the most suitable installation and deployment strategy, we consider several functional requirements that are often encountered. We compare diverse operational environments and provide guidance about which scenario might best serve the needs of an organization. We describe how a high-availability solution architecture can be created either by using the traditional software installation options available or by using the new virtual appliance for IBM Security Access Manager for Enterprise Single Sign-On.

We also illustrate how an organization can manage user expectations by opting for a controlled approach to rolling out the technology incrementally to various systems and users. We then talk describe several of the policy elements linked to the creation of AccessProfiles and how these AccessProfiles can be used to gradually increase the number of applications that an organization wants to enable for single sign-on (SSO).

The chapter concludes by explaining how multi-factor authentication can be used to enforce the security policy of an organization with reduced cost and how audit report generation can help in demonstrating compliancy with regulatory requirements.

3.1 Business requirements

This section describes some of the key questions an organization seeks to answer before designing a single sign-on solution. The outcome of this exercise determines how the single sign-on technology fulfills the business requirements. The installation and deployment methodology that is selected depends on these business requirements and on the functional requirements demanded from the selected technology.

One of the main drivers behind the implementation of a single sign-on solution is the desire to increase security while effectively managing operational IT costs, improving productivity, and addressing compliance concerns.

3.1.1 Increasing security

Typically, organizations seek to implement single sign-on solutions when new and more secure authentication standards are introduced. These solutions can range from mandating more complex passwords to insisting on multi-factor authentication. A risk assessment often precedes the introduction of a new standard.

Organizations analyze their data sets and the applications and systems that interact with these data sets. They determine which of the data sets are most valuable to them and take the steps necessary to reduce the risks associated with them to a level that the organization deems acceptable.

The three security aspects that are typically observed are the confidentiality, integrity, and availability of systems. Revenue producing services need to be able to continue to run. Private data and intellectual property need to be shielded from the outside world. And, the data that is used as the basis of business decisions must be trustworthy.

After determining which risks the organization faces, a security policy is created to reduce the risk level to an acceptable level. One of the mitigating controls that is often selected to reduce risk is to introduce a trustworthy authentication and authorization mechanism for various information technology systems.

Authorization mechanisms prevent unauthorized users from accessing systems, applications, and data. Additionally, when changes are made or when systems are accessed by an authorized user, an audit trail can be generated based on the identity that was used in the authentication process. Changes and data disclosure can be traced back to a specific timestamp and to a specific user identity.

The identities (such as a unique user name), which are used by the authorization process to determine whether specific actions are permitted for a user, are provided by the authentication process.

Authentication is the process where physical persons prove that *they are who they claim to be* to IT systems. The most common way of authenticating to systems and applications is by use of a secret password or personal identification number (PIN) code. Because the password is known by one person only, whoever can provide the correct password is trusted by the application to *be who they claim to be*.

To prevent false identity claims, a password or PIN must be complex. Banks likely do not accept `1234` as an appropriate PIN, and organizations likely do not accept words that appear in a dictionary as acceptable passwords. These passwords are too trivial to guess in a brute-force guessing attack.

Alternatives to passwords: A security policy sometimes insists on stronger authentication methods than password-based authentication. Often a combination of multiple authentication factors is used. Authentication then requires a combination of elements that a person *has* (for example, a smart card or access badge), *knows* (for example, a password) and *is* (for example, fingerprint scan). IBM Security Access Manager for Enterprise Single Sign-On supports multi-factor authentication mechanisms.

Organizations typically use a wide array of applications and systems that can all require various password complexities (dependent on the sensitivity of their nature and the way the application was programmed). This approach can lead to a situation where individuals are forced to create and remember a multitude of user names and passwords. This situation in itself can introduce a new security risk.

As the number of different user names and passwords that individuals are required to remember increases, the likelihood that they can truly keep them safe diminishes. A classic password-coping strategy for many people is to write down the passwords and store them in a restricted environment, such as a private notebook or office drawer. Another coping strategy is resorting to recycling or reusing passwords between different secure and less secure systems and applications. This approach, too, introduces additional risk. The compromise of an outdated insecure application means that the password for more secure systems might also be revealed.

To eliminate the risk of users no longer truly keeping their passwords secret, the implementation of a single sign-on solution is often necessary. A secure and permanently available software *Wallet* allows the users in an organization to automate the process of entering passwords with varying complexity and

expiration rules in the relevant applications. A user needs to provide only one set of credentials to have the entire set of user names and passwords at their disposal.

3.1.2 Reducing costs and improving productivity

The cost that is associated with user authentication can be threefold:

- The development cost of updating or changing applications to allow the use of more secure authentication methods
- The cost related to help desk interventions related to passwords issues
- The loss of productivity of requiring users to log in manually to a multitude of applications and systems

All organizations use a wide range of applications and systems, either developed in-house or sourced from external software vendors. Not all applications and systems have the same type of strong authentication built-in. For several applications, the cost to change the authentication mechanism to a level that meets the chosen standard exceeds acceptable levels. For some end-of-life applications, there is no choice other than to accept an insecure authentication mechanism that was included by the software creators at the time of release.

If an organization wants to enforce a password with several complexity requirements or even insists on the use of multi-factor authentication, but the application has no built-in support for these demands, a single sign-on solution can serve this purpose without the costly process of changing those individual applications.

Another cost factor related to authentication is the user support cost. A significant part of the incidents raised within the average organization is related to password issues. The issues range from people simply forgetting their passwords for infrequently used applications to users that change their password while having caps lock turned on and do not realize it until it is too late. Even just trying to come up with a suitably complex password can be a challenge for many people, forcing them to call on support services.

A need to reduce support costs is the second major driver behind the implementation effort of an enterprise single sign-on solution. IBM Security Access Manager for Enterprise Single Sign-On not only eliminates the need for users to remember a multitude of credential sets, it can also be configured to allow users to reset their passwords themselves through a challenge-response system. It can be linked to generating and provisioning sufficiently complex passwords in a way that is not apparent to the user.

Implementing an enterprise single sign-on solution is also often considered a quick win for the IT department. The productivity gain for office workers throughout the organization is often visible and highly appreciated throughout an organization. Alleviating the hassle for so many people of having to constantly enter user names and passwords manually in various applications over and over again by implementing single sign-on is a visible IT project that yields significant goodwill in other departments.

3.1.3 Addressing compliance

A third requirement that makes the business case for implementing single sign-on technology is the need to be compliant with various regulatory compliance standards and to demonstrate the compliance to auditors.

Many regulation standards require that users sign on to access information with their own credentials. Shared accounts that cannot be individually traced back are often no longer tolerated. Additionally, most standards insist that all access is logged, and that applications must log out if there is inactivity. All of these steps are an effort to minimize unauthorized access to relevant data.

In 3.1.1, “Increasing security” on page 52, we explained how IBM Security Access Manager for Enterprise Single Sign-On can be used to strengthen information access by enabling organizations to enforce stronger passwords and multi-factor authentication. In Chapter 7, “Strong authentication using RFID” on page 227, we also show how IBM Security Access Manager for Enterprise Single Sign-On can be configured so that computers that are left unattended are automatically logged out.

In 9.6, “Tivoli Common Reporting” on page 299, we elaborate on the options that exist to generate on-demand or scheduled audit reports. IBM Security Access Manager for Enterprise Single Sign-On includes built-in user-centric event capture. These event logs can be processed and used in various types of audit reports.

Virtual Appliance: The reporting engine is fully integrated in the Virtual Appliance version of IBM Security Access Manager for Enterprise Single Sign-On. For software installations of IBM Security Access Manager for Enterprise Single Sign-On, you must install the reporting software separately.

3.2 Functional requirements

After the business requirements that drive the need for the enterprise single sign-on technology are defined, the next step is to determine the functional requirements for the technology. The enterprise single sign-on technology can alleviate many of the secure authentication issues that an organization faces. The technology itself needs to be inherently secure. In 3.2.2, “Operational security requirements” on page 61, we provide an overview of several available techniques to IBM Security Access Manager for Enterprise Single Sign-On 8.2 to provide the necessary level of trust to an organization.

Other functional requirements can be specific to every organization. They can, for instance, include a need to support both private workstation sessions and support roaming desktops in virtualized environments. The supported types of operation are introduced in “Session management” on page 31. And, in this chapter, we compare the functionality offered by the available session models.

Another frequently listed demand is the ability to use existing infrastructure when deploying single sign-on solutions. Many organizations, for instance, want to maximize the use of their existing directory and database systems for their single sign-on deployment. This capability bundled with an explanation about how all components can be designed in a high-availability (HA) setup is discussed in 3.2.3, “High-availability design” on page 66. Although high availability typically falls into the category of non-functional requirements, we cover the topic here.

For the next important functional requirement, we elaborate on the various authentication methods. The available second-factor authentication methods are introduced in “Second authentication factors” on page 21. In 3.2.4, “Multiple factor authentication” on page 71, we provide scenarios that illustrate the possible deployment options of these technologies.

There are two final functional requirements that are worth mentioning, although we do not describe them in detail. It is important to avoid changes to the infrastructure, because no client or server needs to be reconfigured to work with IBM Security Access Manager for Enterprise Single Sign-On and no integration work is required for applications. The solution must be extendable and able to insert plug-ins (scripts) to enforce additional authorization checks, to automate certain workflow actions, and to submit custom audit logs.

3.2.1 Comparing the various session management models

IBM Security Access Manager for Enterprise Single Sign-On can be deployed on personal workstations or shared workstations; in private desktops or shared desktops; on Microsoft Remote desktops; on Citrix XenApp Servers, or on

VMware Virtual Desktop Infrastructure (VDI). In “Session management” on page 31, we provide an overview of the characteristics of these various session management models.

Some organizations, however, require a mix of these models when they list their functional requirements. Most organizations have a large pool of workstations that are used by a single user and for which the personal workstation session model suffices. However, the trend to share environments and save on license costs and other resources clearly mandates the use of the *shared desktop* and *roaming desktop* models available with IBM Security Access Manager for Enterprise Single Sign-On.

Table 3-1 provides an overview of the available configuration modes and their advantages and disadvantages.

Table 3-1 Comparison of different desktop operating modes

AccessAgent configuration	Advantage	Disadvantage
► Personal workstation ► One PC, one user, standard SSO	► Easy integration ► SSO to every application possible	► Only basic SSO
► Shared workstation, shared desktop ► One PC, many users, only one desktop	► Easy user switching ► Applications can be kept running under a generic Windows user and be accessed by the different users that log in	► Application context gets lost during user switching, applications that are run as a logged-on user need to be restarted after a user switch
► Shared workstation, private desktop ► One PC, many users, many desktops	► Applications remain open even during user switching ► Fast switching possible	► Not all applications support more than one instance and must be closed during a user switch ► All applications stay in memory, which can consume significant memory
► Roaming desktop	► User can move a session from one desktop to another	► Requires Windows Terminal services, Citrix Presentation Server, or VMware VDI deployment

Dealing with production control applications: Production control applications that take a significant amount of time to calibrate at start-up can be kept running under a generic Windows user. Then, they can be shared between various users in the shared desktop model. Other applications that require individual accountability are logged off and restarted whenever user switching occurs.

Shared desktops

Shared desktops are one of the supported *shared workstation* modes. In a shared desktop mode, multiple users share a generic Windows desktop in one workstation.

After configuring the necessary Windows settings, you must configure IBM Security Access Manager for Enterprise Single Sign-On shared desktop settings in AccessAdmin. You can use the Setup Assistant for this configuration.

You must decide on these policies:

- Who can unlock a computer when it is locked by a logged-on user?
- What must happen at Windows start-up?
- Are users allowed to log off, and if so, what happens then?

Private desktops

Private desktop is another one of the supported *shared workstation* modes. In a private desktop mode, users have their own Windows desktop in a workstation. The private desktop is only visible to the individual user. No other user can view it. If the user copies anything into the clipboard from one desktop, the user cannot paste it into another desktop.

You must decide on these policies to correctly set up the private desktop settings in AccessAdmin:

- What is the maximum number of concurrent user sessions on a workstation?
- What applications must be shared among the various users that log on?
- What happens if someone tries to start a second instance of an application that is shared across desktops?
- What happens if a desktop is inactive for an extensive period?

For more information about these policies, see *IBM Security Access Manager for Enterprise Single Sign-On Version 8.2 Policies Definition Guide*, SC23-9694-01:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itamesso.doc/concepts/ic_policies_guide.html

Roaming desktops

IBM Security Access Manager for Enterprise Single Sign-On supports single sign-on and authentication services for applications hosted on Citrix XenApp Servers, Microsoft Terminal (Remote Desktop) Server, or VMware VDI.

Understanding the difference in the concept: The original IBM Security Access Manager for Enterprise Single Sign-On roaming desktop concept applies to Citrix XenApp and Microsoft Terminal (Remote Desktop) Server only. In these cases, an AccessAgent runs on Windows Server, which supports multiple concurrent user sessions. This concept also applies to the standard and lightweight modes for the Server AccessAgent.

VMware VDI implements a new concept that supports the “roaming desktop” use scenario, where users can access the same desktop wherever they roam. However, it does not involve the use of a Server AccessAgent and its standard and lightweight mode (and virtual channels). The VDI virtual desktop runs a Windows client OS, and AccessAgent is installed in regular client mode there, which runs separately from AccessAgent (if any) at the physical client.

While you install AccessAgent in a Citrix or Terminal Server, which then creates a separate AccessAgent session for each user terminal session, you do not install AccessAgent into a VDI Server.

Instead, you typically pre-install AccessAgent into the golden image (Windows XP or Windows 7) on the VDI system, and the VDI system instantiates one or more virtual desktops for users based off this golden image. In this case, each virtual desktop has its own installed instance of AccessAgent.

You must install AccessAgent on each Citrix or Terminal Server. For every remote session on a Citrix or Terminal Server, an AccessAgent instance runs to help users sign on once (single sign-on) to their applications on the particular remote session. Users can later connect to the same remote session on the Citrix or Terminal Server through any client computer.

AccessAgent installed on a Citrix or Terminal Server can run in *standard mode* or *lightweight mode*. Running in lightweight mode can reduce the memory footprint of AccessAgent on a Citrix or Terminal Server and can improve the single sign-on start-up duration.

See Table 3-2 on page 60 for a comparison of the AccessAgent modes.

Table 3-2 AccessAgent modes

Features	Standard mode (without virtual channel)	Standard mode (with virtual channel)	Lightweight mode
Performance	Normal	Normal	Better
User experience	Log on to Server AccessAgent through EnGINA or Network Provider	Automatic logon to Server AccessAgent with Client AccessAgent credentials	Automatic logon to Server AccessAgent with Client AccessAgent credentials
Supported authentication factors	Not applicable	Radio frequency identification (RFID) Fingerprint biometrics	All authentication factors
Synchronize changes between Client AccessAgent and Server AccessAgent	No, both AccessAgents synchronize separately with the IMS Server in this case	Yes	Yes
AccessAgent Wallet cached on the Server	Yes	Yes	Never
Behavior of AccessAgent when users log on and log off	Depends on graphical identification and authentication (GINA) or Network Provider	Log off remote AccessAgent and disconnect remote session	Disconnect remote session

The single sign-on experience on a Citrix or Terminal Server varies, depending on how you deploy AccessAgent and on the policies that you configure.

The following setup options are available:

- Install AccessAgent on the Citrix or Terminal Server only (Server AccessAgent)
- Install AccessAgent on the client computer (Client AccessAgent) and on the Citrix or Terminal Server
- Install AccessAgent on the client computer (Client AccessAgent) and on the Citrix or Terminal Server but use *lightweight mode* (with virtual channel)

After an organization determines its functional requirements, the organization can customize the Windows desktops or application logon experience hosted on the Citrix or Terminal Servers through policy settings.

For more details, see the *IBM Security Access Manager for Enterprise Single Sign-On Version 8.2 Planning and Deployment Guide*, SC23-9952-03:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itamesso.doc/concepts/ic_deployment_guide.html

3.2.2 Operational security requirements

To better understand how IBM Security Access Manager for Enterprise Single Sign-On implements operational security, we first identify which information assets and procedures must be secured within the software solution.

IBM Security Access Manager for Enterprise Single Sign-On handles the following types of sensitive data:

- Application credentials

  These credentials are stored on behalf of a user to provide automated access to enterprise applications.

- Encryption keys

  These cryptographic keys are used to protect the user credentials.

- Authentication factors

  This secret data that is provided by a user proves the identity of the user to the system. This secret data includes the user IBM Security Access Manager for Enterprise Single Sign-On password, biometric data, and one-time passwords (OTPs).

- Audit logs

  Audit logs must be protected against tampering.

All the sensitive data items listed must be protected as they flow through the system. Thus, we specify the following security requirements for IBM Security Access Manager for Enterprise Single Sign-On:

- Secure storage

  If sensitive data has to be stored, either on the server or the clients, it must be stored in an encrypted form.

- Secure processing

  Sensitive data must be in an unencrypted form while it is used. The system must prevent other user programs from accessing the unencrypted data while it is held in memory.

- Secure communication

  Sensitive data must be protected from eavesdroppers as it travels between the components.

Securing Wallets

In this section, we explain how IBM Security Access Manager for Enterprise Single Sign-On protects all sensitive data items in the components.

Secure storage

Sensitive data that belongs to the users is encrypted with a random cryptographic key that, in turn, is protected by the IBM Security Access Manager for Enterprise Single Sign-On password and other authentication factors of the users:

- Secure storage on the server

  The IMS Server (introduced in 2.2.3, “IMS Server” on page 33) stores only the encrypted forms of the user credentials in its database, so breaking into the database does not reveal the credentials. Moreover, the access controls on the database are configured in so that only an IMS Server-specific database account and the database administrators are granted access to the data.

- Secure storage on the clients

  On client workstations, AccessAgent (introduced in 2.2.1, “AccessAgent” on page 17) stores a copy of the encrypted credentials and common symmetric key (CSK) in a secure data file called *Cryptobox*. Data is stored in an encrypted format by using encryption keys derived from the authentication factors of the users and, optionally, certain machine attributes. AccessAgent can be configured to delete Cryptoboxes if they are not used for a specified number of days.

This approach can minimize the risk of exposure to brute-force attacks on user credentials stored in Cryptoboxes.

Secure processing

AccessAgent protects sensitive data while the data is in the computer memory. The credentials of the users are decrypted on demand and are held in memory temporarily while they are used, and wiped clean immediately after use.

Secure communication

AccessAgent communicates with the IMS Server during logon, periodic sync, and various other scenarios, during which user sensitive data might be transmitted. The communication channel that carries this sensitive data is protected by using Secure Sockets Layer (SSL). After AccessAgent verifies the SSL certificate issued to the server, the communication is encrypted by using

temporary session keys. This approach prevents eavesdroppers from extracting the sensitive data from network packets.

Important: The default expiration of an SSL certificate is one year. Operational teams that manage the IBM Security Access Manager for Enterprise Single Sign-On environment need to include certificate renewal in their planning. The following procedure explains how to renew an SSL certificate:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itamesso.doc/tasks/renewing_ssl.html

Secure audit logs

AccessAgent submits audit logs that contain logon and SSO events to the IMS Server immediately if a network connection exists. Otherwise, the logs are cached offline in an encrypted form and submitted to the IMS Server when the AccessAgent session of the users next has connectivity.

The IMS Server stores audit logs in a set of audit log tables in the IMS database that are protected by DB access control lists (ACLs). These logs can be exported and backed up for safekeeping. Alternatively, the IMS Server can be configured to forward logs to a designated SysLog server, where they can be integrated with a centralized security *information and event management system*.

Recovering Wallets

The Wallet of a user is protected by an encryption key, which, in turn, is protected by the IBM Security Access Manager for Enterprise Single Sign-On password. If the user forgets the password, the credentials stored in the Wallet are not available, which prevents the user from accessing enterprise applications. IBM Security Access Manager for Enterprise Single Sign-On provides the user with various means to recover the Wallet, even if the password is forgotten. During registration, a user is allowed to register one or more personal secrets. These secrets are responses to questions only the user is likely to know. If the user forgets the password, the user must provide a specified number of correct personal secrets to reset the password and recover the Wallet. In this process, IBM Security Access Manager for Enterprise Single Sign-On re-encrypts the user CSK with the new password provided by the user.

For more details about this topic, see Chapter 6, “Password self-services implementation” on page 199.

Strengthening the protection of Wallets

Because logging on to IBM Security Access Manager for Enterprise Single Sign-On provides the user with the credentials to log on to multiple enterprise

applications, the authentication to IBM Security Access Manager for Enterprise Single Sign-On must be strengthened. IBM Security Access Manager for Enterprise Single Sign-On provides several ways to strengthen the authentication.

Use of password policies

For Active Directory-based deployments, the IBM Security Access Manager for Enterprise Single Sign-On password is synchronized with a user's Active Directory password and is subject to the same password policies set by the organization's Active Directory.

For non-Active Directory deployments, the organization can configure IBM Security Access Manager for Enterprise Single Sign-On password policies to ensure that *strong passwords* are used.

Use of authentication factors

Access to the Wallet can also be strengthened by enforcing the use of additional authentication factors, such as RFID badges, finger biometrics, and smart card tokens (see also 2.5.9, "Third-party readers" on page 49).

The use of multi-factor authentication increases security. An attacker now needs to obtain both a physical token and a password or a PIN of a user to gain access to a Wallet. IBM Security Access Manager for Enterprise Single Sign-On can use RFID-enabled facility access badges as authentication factors. Users must present their RFID access badge and the password to log on to their systems. To log on using a smart card token, the users supply the smart card PIN, which is verified by the smart card itself. The private data on the smart card is protected by the PIN, which is locked out after a pre-configured number of successive failed attempts. IBM Security Access Manager for Enterprise Single Sign-On uses Public Key Cryptography to authenticate the smart cards to the IMS Server by using 2048-bit Rivest-Shamir-Adleman algorithm (RSA) keypairs stored on the smart cards.

Finger biometrics can be used for authentication without the use of a password. To log on, a user places the appropriate finger on a fingerprint reader. A template of the finger minutia is created and compared with the template of that user's finger created at initial enrollment. The fingerprint itself is not stored and the template cannot be reverse-engineered to replicate the fingerprint, providing an additional layer of security.

User convenience: As an added convenience, policies are often designed so that they allow cached passwords for several hours when using an RFID access badge, for instance. Users are then only asked to present both their RFID badge and enter their password one time every *n* hours.

How the private desktop feature ensures security

The private desktop feature is provided by AccessAgent. It uses the Windows operating system support to create multiple Windows desktops for different user accounts. It uses the user's own Windows privileges and facilitates the switching between these desktops. This way, the private desktop is only visible to the individual user; no other user (including the administrator) can access it.

In Windows XP, when a new user logs on from the AccessAgent graphical identification and authentication (GINA), the private desktop first verifies that the user is a valid user, and then creates a Windows desktop for that user. It then loads the Windows profile of the user, and creates the shell of the user (starting Windows Explorer, and so on) for the user to interact with the desktop. The private desktop also provides Global Policy Object (GPO) support by starting the client-side extensions to apply the group policies applicable to the user. Next, the user shell in the security context of the user is created. Therefore, all applications run from the desktop are executed in the user's own security context.

In Windows 7, IBM Security Access Manager for Enterprise Single Sign-On uses Windows *fast user switching* (FUS) to support concurrent multiple user sessions on the same workstation. It uses the security features built into Windows FUS. IBM Security Access Manager for Enterprise Single Sign-On enhances the native FUS functionality. It enforces IBM Security Access Manager for Enterprise Single Sign-On authentication policies (for example, second-factor authentication) before granting user access into private desktop sessions on the workstation.

With the private desktop session, each desktop runs with the rights of the user's Active Directory account, and so, access to each user's desktop/resources remains protected by Windows access control. As long as each user account does not have administrative rights on the machine, a user cannot possibly access another user's data.

When users log off from their desktop, the private desktop gracefully logs off the users' applications by sending end session messages to each open window on the users' desktops. Similar to a normal Windows logoff when an application is not ready to end, the private desktop displays a notification to the users and lets them terminate the logoff processes. If a system restarts or shuts down, all private desktops are logged off gracefully before the system restarts or shuts down.

The private desktop is designed to prevent malicious software or some other desktop management software from switching between a current desktop to another user's desktop. If a third-party software tries to perform desktop switching, AccessAgent immediately locks the workstation. If the component of

AccessAgent that implements this security measure is somehow terminated by the administrator, the computer is restarted automatically.

This functionality also prevents the clipboard content on one desktop from being accessed from another desktop session. Anything copied onto the clipboard from one desktop is prevented from being pasted into another desktop.

3.2.3 High-availability design

Implementing high availability is about ensuring that services are always available. Several layers of redundancy are built into IBM Security Access Manager for Enterprise Single Sign-On. Whether an organization wants to implement high availability for every component of the single sign-on architecture is part of the organization's functional requirement assessment. Some organizations insist on implementing high availability for every component. Other organizations choose to rely on the caching functionality offered by AccessAgent and a disaster recovery procedure (as explained in "Disaster recovery" on page 71) to cope with a potential failure of the IMS Server.

The second option of not implementing high availability at all component levels, and instead relying on a tested disaster recovery procedure, can be a valid option[1]:

- If the IMS Server is not available, AccessAgent can remain functional, because AccessAgent caches system data into a machine Wallet and user data into individual user cached Wallets.
- When the server is offline, AccessAgent can continue to authenticate users with one or two authentication factors by using the authentication data cached on the local computer.
- AccessAgent can provide single sign-on for the user when the server is offline by using the cached user Wallet.
- If the user forgets the password or second authentication factor, IBM Security Access Manager for Enterprise Single Sign-On provides various ways for users to regain access to the user Wallet.

In Chapter 2, "Single sign-on architecture and component design" on page 11, we explain the logical components of the IBM Security Access Manager for Enterprise Single Sign-On solution. A simplified architectural overview of these components is shown in Figure 3-1 on page 67.

[1] A risk assessment must determine whether high availability is a functional requirement for IBM Security Access Manager for Enterprise Single Sign-On components.

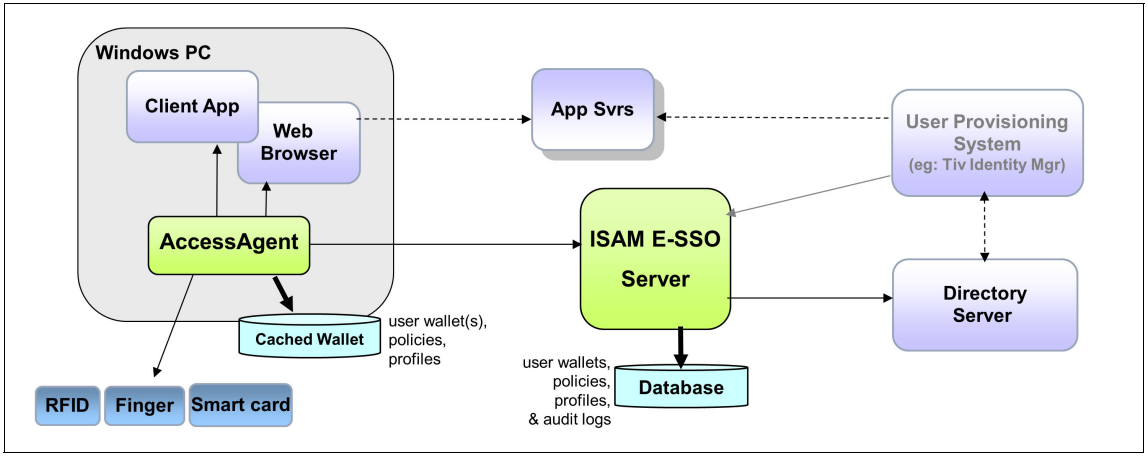

Figure 3-1 Logical relationships between client and server components

All components can be configured in a high-availability mode. We first briefly include a description of database and directory servers. We then focus on the actual single sign-on server components, whether it is a virtual appliance or a traditional software installation on Windows servers.

Database high availability

IBM Security Access Manager for Enterprise Single Sign-On uses the existing high-availability features of the database products with which it works.

The most popular method for database HA is the use of clustering features supported by the database product. Typically, this method is used in conjunction with a compatible third-party HA solution, such as Tivoli System Automation for Linux/AIX or Microsoft Cluster Server (MSCS) for Windows. A typical clustered configuration involves an active-passive pair of database nodes with access to shared disk storage (for example, a storage area network (SAN)). The IMS Server naturally works with a clustered database, because the cluster service presents a single database endpoint to which the IMS Server connects.

There are alternative methods of maintaining an active-passive pair of database nodes, without the requirement of shared disks, such as the DB2 high availability disaster recovery (HADR) feature. These DB HA features might not necessarily support automatic failover in its basic configuration, and typically require additional software or hardware to support automatic failover. For example, DB2 HADR requires an initiator function, such as Tivoli System Automation, to support automated failover. Additional database HA configurations might require that a special Java Database Connectivity (JDBC) connection string, for example, which identifies all the database nodes, is configured at the IMS Server

WebSphere Application Server tier. This JDBC configuration is not supported by the IMS configuration utility. The administrator needs to manually configure the data source at WebSphere Application Server for the IMS Server to use.

Supported databases: IBM Security Access Manager for Enterprise Single Sign-On supports IBM DB2, Microsoft SQL, and Oracle databases.

Directory server high availability

IBM Security Access Manager for Enterprise Single Sign-On does not store any data on the enterprise directory, does not require any directory schema extensions, and does not connect to the directory server for most single sign-on scenarios.

IBM Security Access Manager for Enterprise Single Sign-On relies on the directory server to verify user identities during sign-up. If password synchronization is configured, IBM Security Access Manager for Enterprise Single Sign-On also connects to the directory server when performing password reset and password synchronization tasks.

WebSphere configuration to enable directory server availability: To ensure high availability at this level, configure the Virtual Member Manager (VMM) component of the WebSphere Application Server (introduced in 2.2, “Logical component architecture” on page 16) to communicate to *more than one* Active Directory domain controller instead of a specific domain controller.

We can configure VMM with just an Active Directory domain name (so that it can locate any Active Directory controller) only if secure Lightweight Directory Access Protocol (LDAPS) is enabled on Active Directory.

If LDAPS is not enabled on Active Directory, you need to configure VMM with the host name of multiple individual Active Directory controllers for failover purposes.

Virtual appliance replication for high availability

An organization can choose to deploy a ready-to-run server virtual appliance on VMware ESX or ESXi for faster deployments, as we describe in 3.3, “Deployment strategies” on page 75. These server virtual appliances can be deployed in a stand-alone or high availability configuration, as shown in Figure 3-2 on page 69.

To achieve high availability for the IBM Security Access Manager for Enterprise Single Sign-On server components when you use virtual appliances, use the Export and Import configuration tools.

The Export and Import configuration tools can be used to replicate the IMS Server configurations among virtual appliances. You can deploy and activate two or more virtual appliances. After a virtual appliance is configured completely, you can export the configuration to another deployed virtual appliance. Figure 3-2 shows two virtual appliance replicas that are configured the same for high availability with a load balancer in front of them.

No integrated database: The virtual image does not have a preinstalled database server. An external database is still required, and it needs to be separately configured for high availability.

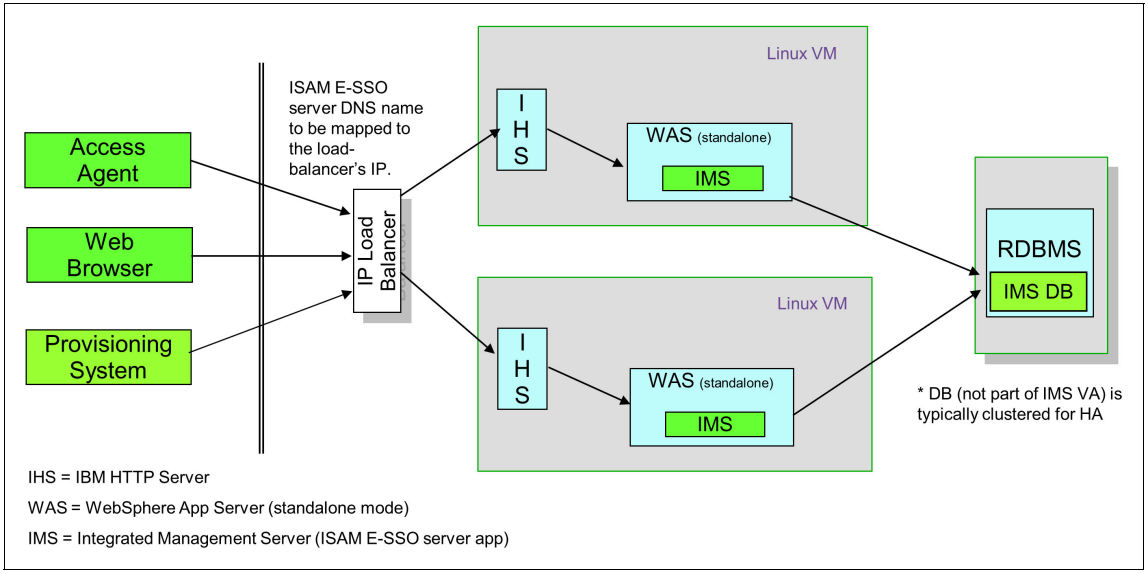

Figure 3-2 Two virtual appliance replicas configured the same for high availability with a load balancer

In the configuration that is shown in Figure 3-2, there is no reliance on the IBM HTTP Server to balance the load across the WebSphere Application Server. Each IBM HTTP Server connects to the WebSphere Application Server instance on the same virtual appliance only.

Load balancing and clustering

An organization can also achieve high availability for the IMS Server by setting up multiple hosts with the software installations of the IMS Server on their own Windows servers if they prefer. They can use a load balancer as the deployment front end with session-awareness and *automatic failover* capabilities.

The IMS Server architecture consists of multiple tiers, as shown in Figure 3-3:

- Load balancer
- Web server tier (IBM HTTP Server)
- Application tier (WebSphere Application Server Network Deployment)
- Data tier (DB2, Oracle, or Microsoft SQL Server)

Figure 3-3 shows a multiple tiered deployment with a load-balancing IP infrastructure as the deployment front end to distribute AccessAgent client requests.

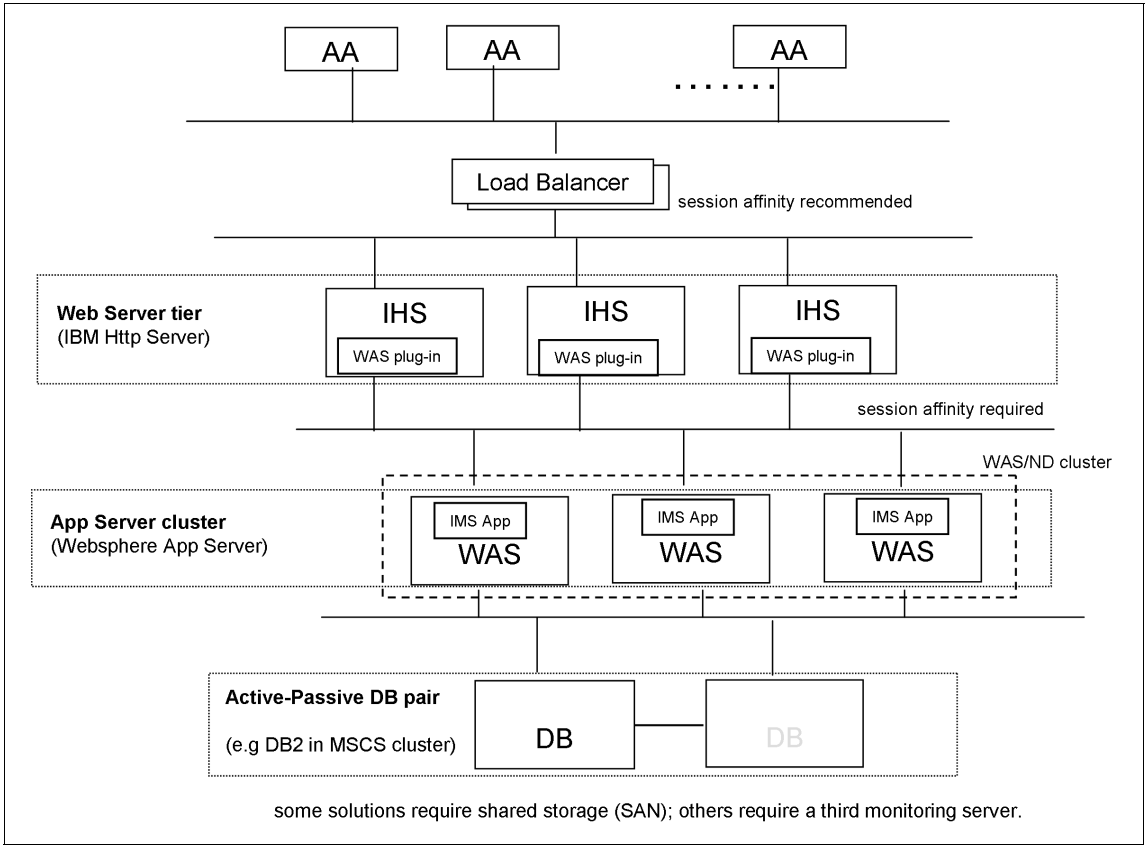

Figure 3-3 Multitiered deployment with a load-balancing IP infrastructure

In Figure 3-3, the load balancer routes traffic to the web server tier, which in turn routes traffic to the application server tier. The load balancer is responsible for distributing incoming requests evenly to a collection of IMS Servers on the application tier. By using a load balancer with session affinity, traffic from each client is always routed to the same IBM HTTP Server.

In this configuration, it does not matter whether the IBM HTTP Server is on the same host as the WebSphere Application Server. Each IBM HTTP Server can send requests to any WebSphere Application Server instances, on the same or a different host.

Disaster recovery

An organization can also decide to set up disaster recovery (DR) for IBM Security Access Manager for Enterprise Single Sign-On. The organization sets up a standby of the IMS Server and its database at a designated disaster recovery site.

Considerations for disaster recovery design differ from availability. Disaster recovery focuses on the actions and processes to recover from a disaster that strikes the existing infrastructure.

To set up disaster recovery, an organization must configure the IMS Server to use the same configuration as the active IMS Server, except for the IP address and the IMS Server database.

The organization can use the Export and Import configuration tools to copy the production environment configuration and replicate it in the disaster recovery environment.

Mirroring the database to a DR site: The standby database can be kept updated through log shipping or database mirroring technologies of the database vendors, for example, DB2 HADR.

After the standby database is set up, you can switch to the disaster recovery site by using one of the following methods:

- Switching the standby database to active mode so that AccessAgent is redirected to synchronize with the IMS Servers at the disaster recovery site
- Starting the IMS Server service on the standby server hosts
- Switching the Domain Name System (DNS) settings so that AccessAgent points to the IMS Server at the disaster recovery site

3.2.4 Multiple factor authentication

Authentication factors are available in different forms and functions, such as passwords and devices that work like a key. A strong authentication setup reduces the risk of security compromises. Organizations can implement a second authentication factor or an alternative authentication factor to secure user sessions.

These more secure authentication standards are not demanded for all applications. These more secure authentication standards are required for those applications where the importance of the data or the likelihood of an attack is too great to rely on passwords alone.

IBM Security Access Manager for Enterprise Single Sign-On supports the use of different authentication factors, such as fingerprints, RFID and smart card devices, and one-time passwords (OTPs).

The devices or codes can be used as second authentication factors. You can also use a fingerprint reader as an alternative authentication factor to a password.

Authentication policies can vary per user group and per machine group. For example, you can roll out RFID cards to users in one department, smart cards to another group of users, and password-only authentication for a third group of users. You can configure some systems with RFID readers and other systems with fingerprint readers. You can allow users to register more than one second factor, such as an RFID card and smart card, or to easily switch from one second factor to another factor. Users can have multiple authentication factors registered.

The factors depend on the analysis of the functional requirements created by the relevant departments and on the security policy and standards defined by the entire organization.

Table 3-3 on page 73 lists examples of a use-case scenario and a particular target group for each authentication factor.

Table 3-3 Authentication factor scenario examples

Authentication factor	Use case scenario	Targeted user group
Building access badge	Users tap their building access badge on the reader and enter a password to log in to IBM Security Access Manager for Enterprise Single Sign-On and gain access to the network or applications.	This solution is best suited for business users that work within the corporate premises.
Active RFID	Users are identified as they approach the workstation. They simply enter a password to log on.	This solution is best suited for staff members that require fast logons and logouts. Unlike building access badges, users do not have to tap their badges on the readers to identify themselves. This solution is more convenient, but it also has a higher total cost of ownership.
Mobile device	Users receive a Mobile ActiveCode on their Short Message Service (SMS)-enabled or email-enabled mobile device. Users use this code with their IBM Security Access Manager for Enterprise Single Sign-On user name and password to log on. This factor works in an AccessAssistant and Web Workplace environment.	This solution is best suited for remote users who need a second factor to log on remotely. This solution frees the remote users from having to carry additional devices while still maintaining access security. Mobile ActiveCode is also more cost-efficient compared to traditional one-time password tokens.

Authentication factor	Use case scenario	Targeted user group
iTag (identity tag)	Users use any personal device or photo badge with smart labels or sticker labels to enable two-factor authentication.	This solution is an alternative to conventional token-based two-factor authentication. With iTag, users do not incur the inconvenience of carrying a separate token for authentication. By using what users already have for authentication, enterprises can minimize the cost and complexity of two-factor authentication projects.
Finger biometrics	Users use their fingerprints to log on to IBM Security Access Manager for Enterprise Single Sign-On.	This solution removes the need to enter a password. Unlike some strong authentication methods, such as RFID and smart cards, fingerprints cannot be shared, borrowed, or stolen, so there is an exceptionally high degree of certainty that the user authenticated is the correct person. This solution is an alternative to building access badges or Active RFID badges for clinical staff.
USB smart token	Users insert a USB smart token and enter a password to log on to IBM Security Access Manager for Enterprise Single Sign-On.	This solution is best suited for business users or mobile computer users who require a higher level of security protection.

Authentication factor	Use case scenario	Targeted user group
One-time password tokens	Users carry an authentication token, which is used to generate a one-time password. Users use this password with a pin that they remember to log in to corporate networks. This factor works in an AccessAssistant and Web Workplace environment.	This solution is best suited for remote users who require a second factor to log in remotely to corporate networks.
Strong passwords	Users enter an IBM Security Access Manager for Enterprise Single Sign-On user name and a strong password to log on.	This solution is suited for any group whose risk profiles do not warrant a second factor, or where a second factor is not viable.

Multiple factor authentication is deployed as a security enhancement, but also for user convenience. IBM Security Access Manager for Enterprise Single Sign-On allows policies to be defined that allow the caching of credentials for a defined period. Policies can, for instance, be designed that demand two factors to be presented at the start of the work day (password and RFID, for instance). But, then, the policy might allow the users to present their RFID badge only during the following eight hours without entering the password again.

Fast user switching in production environments: Production environments often deploy RFID technology for fast user switching. This process is fast. It can address the case where users wear gloves, where reducing the number of keyboard interactions can be a valid requirement.

For more details about how to enable the various second-factor authentication modes, see the *IBM Security Access Manager for Enterprise Single Sign-On Version 8.2 Configuration Guide*, GC23-9692-01:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itamesso.doc/concepts/ic_config_guide.html

3.3 Deployment strategies

After the functional requirements are documented, the actual deployment of the single sign-on environment can be planned.

In Chapter 5, “Base installation and configuration” on page 107 of our complete healthcare scenario, we describe how to install and configure the components of IBM Security Access Manager for Enterprise Single Sign-On.

In this section, we provide basic pointers about determining the type of deployment that can be selected. We explain how to control the process to ensure that user expectations are properly managed throughout the process.

3.3.1 Plan for IMS Server scalability

In 3.2.3, “High-availability design” on page 66, we mentioned the various deployment options that exist for IBM Security Access Manager for Enterprise Single Sign-On.

One of the first questions that needs to be answered is what type of deployment is most suitable for an organization. This answer depends on the functional requirements and on the risk tolerance level (regarding high availability). We include several generic preferred practice guidelines that illustrate the various deployment options. These guidelines are based on the number of users of the single sign-on environment. Also, these guidelines are based on the number of applications that are expected to have a defined AccessProfile (to enable them for single sign-on).

Suggestion: Single sign-on projects tend to include more applications as time progresses. Even if an organization is successful at protecting the scope definition of a single sign-on deployment project, it is worthwhile to consider growth in the number of supported application profiles when you determine the most suitable deployment type.

The following deployment options exist for IBM Security Access Manager for Enterprise Single Sign-On:

- For small scale deployments (or proof of concept (POC) setups):
  - Fewer than 10,000 users and fewer than 50 profiles.
  - Virtual appliance is suggested (unless fingerprint is a requirement).
  - Stand-alone deployment if fingerprint is required.
- For medium-scale to large-scale deployments:
  - 10,000 - 50,000 users.
  - Network Deployment (clustered software installation) is suggested. Another option is a stand-alone deployment.

- For large-scale deployment:
  - Typically consists of more than 50,000 users and requires more than 50 profiles.
  - Network Deployment (clustered software installation) is required.

Small-scale deployments that use a virtual appliance

The deployment of a virtual appliance is a new option since the release of IBM Security Access Manager for Enterprise Single Sign-On 8.2. This option uses the deployment of a virtual appliance to manage a single sign-on environment rather than installing various software components on the actual Windows servers.

Virtual appliance support: The IBM Security Access Manager for Enterprise Single Sign-On virtual appliance is designed to run on VMware ESX/ESXi hypervisors only. There is no support to configure it to run on other hypervisors and virtualization solutions.

It is easier to install and configure the IMS Server by using a virtual image that runs on a hypervisor. The virtual image contains a preinstalled WebSphere Application Server, IBM HTTP Server, Tivoli Common Reporting, and IMS Server. You need to deploy, activate, and configure the virtual appliance only. However, an external database is still required.

Important: Fingerprint authentication is not supported on a virtual appliance deployment. If fingerprint authentication is a functional requirement, opt for a software-based installation on Windows servers instead.

Large-scale deployment model

Medium-scale to large-scale architectures with, for example, up to 500,000 users adopt the standard two-tier architecture, with multiple IMS Servers in the front-tier and a clustered IMS database in the back end. See Figure 3-3 on page 70 for a reminder of the architectural overview of this model. It is also described in 3.2.3, “High-availability design” on page 66.

The IMS Servers (software installation on Windows servers) must, in this model, be fronted by a session-aware load-balancer. The IMS tier is thus horizontally scalable. If you assume that a Windows server can support 40,000 concurrent AccessAgent sessions, a deployment that expects 100,000 concurrent AccessAgent sessions requires three servers at the IMS tier to share the load.

The database tier can be scaled vertically only by default. The database host must be sized correctly in relation to the number of IMS Servers that it is expected to service. A typical guideline, based on load test experiments, is to

double the processor and memory capacity of the database host for every five additional IMS Servers.

For a more detailed description of the various deployment options, see the *IBM Security Access Manager for Enterprise Single Sign-On Version 8.2 Planning and Deployment Guide*, SC23-9952-03:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itamesso.doc/concepts/ic_deployment_guide.html

3.3.2 Deployment time estimation

Your estimates for timing depend largely on the following factors:

- Number and types of users and desktops

  The size of the deployment is directly proportional to the number of users and desktops. As the number of users increases, the more the system depends on performance tuning. As part of those numbers, consider the number of total roaming desktop sessions potentially active at any one time. Consider how many total workstations require that an AccessAgent is installed.

- Number and types of applications

  The number of applications plays a lesser role in estimating the size of the deployment than the types of applications. As the number of applications increases, the design and test phases of the deployment increase. However, consider the number of applications that require advanced profiling techniques. Profiling complex applications can increase the overall time of deployment.

- Advanced deployment requirements

  Advanced deployment tasks, such as incorporating an identity management system for provisioning of IBM Security Access Manager for Enterprise Single Sign-On user credentials, and IMS clustering are two examples that increase the overall complexity of the deployment. These tasks and advanced profiling requirements must be factored into the overall time estimations and necessary deployment skills.

3.3.3 Initial deployment scenario

Now, we look more closely at possible initial deployment steps for the IBM Security Access Manager for Enterprise Single Sign-On base components. For a description of the various components, see Chapter 2, “Single sign-on architecture and component design” on page 11.

Follow these steps:

1. The administrator configures a database on the IMS database server. The administrator installs the IMS Server software. This step must be done before any AccessAgents are installed.
2. If the administrator wants to manage SSO profiles from the administrator's system, the AccessAgent and AccessStudio software must be installed.
3. With the help of the IMS Configuration Utility, the administrator configures the IMS Server with the organization directory (Active Directory, for instance) to authenticate the user. At this step, the initial AccessProfiles and machine policies are defined:
   a. Set up the infrastructure (hardware and requisite middleware).
   b. Install IMS into the infrastructure.
   c. Configure IMS (set up server certificates, database tables, and directory connections) by using the IMS Configuration Utility.
   d. Set up IMS policies (by using the AccessAdmin SetupAssistant).
   e. Incrementally create and upload AccessProfiles (by using AccessStudio).

   All configuration items that belong to the profiles, such as AccessProfiles or machine policies, are stored in the IMS database.
4. When the IMS Server is running and the initial profiles are defined, the AccessAgent can be deployed onto the Windows clients. During the installation phase, the AccessAgent registers itself at the IMS Server and downloads the required system and machine policy.

If manual sign-up for new IBM Security Access Manager for Enterprise Single Sign-On users is configured, the user must authenticate with the organization directory credentials. Usually, these credentials are the Active Directory credentials. The IMS Server always checks the sign-up credentials against the organization directory that is configured into the IMS Server.

During the user sign-up, a new Wallet for the user is created, stored in the IMS database, and downloaded to the AccessAgent together with the required UserProfile. From this point forward, the user operates as usual.

During normal operation, the user authenticates against the AccessAgent. The IMS Server proofs the user credentials provided by any authenticator against the IMS Server or against the cached Wallet of the user if the AccessAgent is offline to the IMS. If the authentication is successful, the AccessAgent synchronizes any modified AccessProfiles or UserProfiles from the IMS Server.

3.3.4 Managing expectations

One of the main focus points during the deployment of a single sign-on solution must be the user community. Single sign-on technology can greatly affect the user experience with the use of systems and applications. The transition to using single sign-on is visible and noticeable. It is critical to the success of a single sign-on deployment to always remember the user and actively manage the expectations of the user.

Users must be briefed about what changes to expect after the single sign-on functionality is available to them. They must be briefed about what actions (for instance, signing up) are expected of them, and why and how their life is easier for a predefined set of applications.

When introducing alternative authentication factors (RFID or smart card), it is important to involve the user community even during pilot phases. The feedback of the users can warrant policy changes prior to a first deployment to a larger group of users.

Incremental deployments: We want to emphasize the importance of deploying *incremental rollouts* of second factors and AccessProfiles to machines and users.

Incremental deployments provide additional time for administrators and users to educate and familiarize themselves with the new technology. Incremental deployments provide additional time for issues to be identified and resolved early.

The IBM Security Access Manager for Enterprise Single Sign-On system is designed to avoid an organization that takes a *Big Bang approach* to rolling out the system organization-wide.

User education

Consider how to educate the users about the single sign-on system. Users need to be alerted in advance about the single sign-on system. The approach typically varies from organization to organization. Some organizations train team leaders who pass the information on to their team members. Other organizations post training videos on an internal website. Other organizations use leaflets or booklets. Set the expectations of the users. There might be times when the system does not work as expected. It is also advisable not to launch too many applications at the same time or to change focus during the login process.

Importance of managing expectations: Users sometimes complain if they are not given a perfect system, forgetting that their use of the system is now much easier.

3.3.5 Enabling single sign-on for applications

A successful deployment of IBM Security Access Manager for Enterprise Single Sign-On typically focuses on a select number of applications at launch. An organization selects a few key applications that it wants to tie into the single sign-on client (AccessAgent) and design, test, and deploy a profile for it.

After the first set of AccessProfiles is distributed to the users in the organization and after the user education, systems, or the profiles are tweaked, decide (after a positive evaluation) to expand the scope with additional applications.

Each application is represented by an *AccessProfile*, which is a set of instructions that defines the automatic logon mechanism for that particular application.

Administrators create AccessProfiles for these applications to configure support for additional applications. The AccessAgent then reads the AccessProfiles and performs workflow automation, such as sign-on and sign-off automation.

The complete solution provides these features:

- Automatic application account provisioning
- A central view of all application accounts
- Sign-on and sign-off automation
- Authentication management
- User-centric audit logs and report generation
- Centralized de-provisioning for all accounts

AccessStudio features and benefits

AccessStudio is an IBM Security Access Manager for Enterprise Single Sign-On component, which Administrators use to create and manage AccessProfiles.

AccessStudio provides maximum control over configuring AccessProfiles. AccessStudio provides maximum control over managing Authentication Services and their associated data (which includes application objects, authentication services, authentication service groups, and authentication service group links).

You can set up AccessProfiles for the following types of applications:

- Windows applications
- Web applications

- Applications that use Java applets
- Terminal applications
- Mainframe applications
- Applications with bitmapped windows

AccessStudio offers the following features:

- Standard and advanced modes of AccessProfiles that support requirements of varying complexity
- Multiple editing that uses GUI-based and XML editors
- Flexibility in editing AccessProfiles stored in any location, including those AccessProfiles in the IMS Server
- The ability to import existing AccessProfiles from a local installation of AccessAgent or from the IMS Server
- Automatic validation of user-configured AccessProfile data, to minimize errors
- The ability to test and debug AccessProfiles

AccessProfile testing

An organization can use the Test function of AccessStudio to verify whether the AccessProfiles for their applications function correctly.

When you start your test, launch the applications with the configured AccessProfiles in AccessStudio. The test is executed for all AccessProfiles whose corresponding applications are active on the computer. A log is created for each of these applications in addition to the existing logs.

Important: When you start a test that uses AccessStudio, AccessAgent Wallet is temporarily cleared until the test is stopped. This setting means that logon automation on your computer does not work until the test is stopped.

Supported applications

The AccessStudio wizard auto-generates single sign-on AccessProfiles for a broad range of applications:

- Web applications accessed through the Microsoft Internet Explorer or Mozilla Firefox browser
- 32-bit and 64-bit mainframe applications
- Teletypewriter (tty) applications based on text-out technology, such as PuTTY
- Visual Basic applications
- .Net applications
- Mainframe HLLAPI (login window only)

- Applications with owner-drawn user interfaces (login window only)
- 32-bit Java applications (login window only)
- 32-bit Java applets (browser-based and login window only)

Profiles available for download

AccessProfiles are released for a fixed set of logon work flows for third-party applications. AccessProfiles are released for various languages and environments.

IBM Security Access Manager for Enterprise Single Sign-On provides AccessProfiles on the IBM Support site. They can be downloaded from the AccessProfiles Library at the following location:

http://www-304.ibm.com/support/docview.wss?uid=swg21470500&wv=1

3.4 Log collection and audit reporting

In this section, we explain how the logging functionality offered by IBM Security Access Manager for Enterprise Single Sign-On can be used to feed into security audit reporting. We also describe how the logging functionality can provide business analysts with metrics about user activity on all desktops on which the AccessAgent is deployed.

3.4.1 Audit log collection

IBM Security Access Manager for Enterprise Single Sign-On has a custom audit log[2] action framework that can log any event on the desktop. For example, log an event whenever a user opens a Microsoft Word file. This tracking capability uses the AccessAgent plug-ins platform to automate the collection of custom audit trails at the enterprise endpoints.

You can create custom events to track application-specific events:

- Access to confidential data
- Attempted access to application features that the user is not authorized to use
- Access to applications outside of office hours

IBM Security Access Manager for Enterprise Single Sign-On generates event logs at all endpoints. Administrators and help desk officers can access the audit logs for individual users. Only administrators can run full queries on audit logs,

[2] An audit log displays the details of each activity, for example, user name, date, and the result of the activity.

access the help desk logs, and generate reports about help desk and user activity. Users do not have read or write access to these logs.

Types of logs

There are two types of logs:

- User logs
- Administrator logs (including help desk activities)

IBM Security Access Manager for Enterprise Single Sign-On tracks the following information:

- What applications users access
- Who accessed these applications
- Details about the accounts used
- When users accessed these applications, and from where they are accessed

Storage and sync

If AccessAgent is connected to the IMS Server, AccessAgent audit logs are immediately submitted to the IMS Server. The IMS Server stores the audit logs on a relational database. If there is no network connection to the IMS Server, AccessAgent temporarily caches the event logs on the local computer. The logs are submitted to the IMS Server when network connection to the IMS Server is restored.

3.4.2 Audit reporting

Organizations can use Tivoli Common Reporting to create, customize, and manage audit reports based on the log data collected in IBM Security Access Manager for Enterprise Single Sign-On. Additionally, IMS externalizes a number of database views that can be accessed by an organization to create its own custom reports by using its existing SQL reporting tools.

Report formats: The Tivoli Common Reporting tool can generate reports in HTML, PDF, Microsoft Excel, or Adobe PostScript format.

IMS Server reports are packaged as Business Intelligence and Reporting Tools (BIRT) reports that can be imported into any Tivoli Common Reporting server. Tivoli Common Reporting connects directly to the database. You can use Tivoli Common Reporting to produce reports on the audit events, even if the IMS Server is not running.

There are four reports bundled with IBM Security Access Manager for Enterprise Single Sign-On:

- Application Usage (authentication service activity of one or more users, sorted by event, and time)
- Help Desk Activity (activity of one or more help desk users sorted by event and time)
- Token Information (activity of one or more users sorted by token type, event, and time)
- User Information (activity of one or more users sorted by event, result, and time)

This type of reports is interesting to security officers, administrators, and business analysts. Statistics about which applications actually are used in daily operations can aid in steering future business decisions and investments.

3.5 Conclusion

In this chapter, we explained how business requirements can drive single sign-on implementations and how functional requirements help determine the design of a solution architecture.

Every organization must assess its high-availability needs for the various components and evaluate the benefits that multiple factor authentication can offer. IBM Security Access Manager for Enterprise Single Sign-On offers a way to tailor the solution design to the needs of the organization.

We explained what a typical deployment process can look like and provided advice about how to manage user expectations during an incremental rollout of IBM Security Access Manager for Enterprise Single Sign-On.

We concluded this chapter by describing how audit reports can be generated and managed through Tivoli Common Reporting for IBM Security Access Manager for Enterprise Single Sign-On logging. We described how this logging can be used to fulfill both daily operational security needs and provide business analytics.

Part 2

Customer environment

In this part, we describe how you can use IBM Security Access Manager for Enterprise Single Sign-On in a particular client situation.

4

Overview of scenario, requirements, and approach

This chapter introduces a typical business scenario of a fictional cardio healthcare company, referred to as *the cardio healthcare company* or the *company*. It shows how the company can use the IBM Security Framework and IBM Security Blueprint to help secure and facilitate its user authentication processes.

This chapter includes the following sections:

- Company overview
- Business vision
- Business requirements
- Functional requirements
- Design approach
- Implementation approach

4.1 Company overview

The cardio healthcare company is a healthcare provider that focuses on providing specialized cardiovascular-related healthcare services in the US. The company was founded in California and then expanded across the country. It operates stand-alone clinics in several states, where each clinic occupies its own building and provides preventive care, cardiac surgery, and outpatient services. The team of professionals that work at the company consists of a large group of medical and supporting staff directly employed by the company and a smaller group of independent surgeons contracted by the company.

The cardio healthcare company maintains financial data and private customer data (patients, research partners, and affiliated hospitals). Most records are kept in electronic form in SAP systems. In addition, email is available to the entire staff of the company to communicate internally and with the outside world (patients and external partners).

Planning to use Epic Systems Electronic Health/Medical Records (EHR/EMR): The cardio healthcare company plans to transition in the next year to EHR/EMR (http://www.epic.com/) for their healthcare records management needs. For all medium-term IT projects, support for both SAP and Epic Systems is required. IBM Security Access Manager for Enterprise Single Sign-On provides SAP AccessProfiles and provides an adapter for Epic Systems Electronic Health Records (EHR).

The cardio healthcare company built a strong and long-term reputation and financial stability over the past 15 years in the US. The company plans to expand operations within the US and to open healthcare centers in international markets.

The following section provides an overview of the information technology (IT) infrastructure that supports this business.

Staying focused: The following sections describe company information that is relevant to the security solutions of the People & Identity domain. It does not provide a complete description of the company and it does not address all the necessary activities related to information security.

4.1.1 Current IT infrastructure

The cardio healthcare company relies on two data centers: a *primary site* (in Phoenix, AZ) and a *backup site* (in Raleigh, NC). All production-related operations are in the primary data center. In terms of production, the backup data center is used for disaster recovery (DR) only.

The backup data center is also used for development and quality assurance (QA) tests on the applications and the infrastructure. Most of the business applications are SAP-based. All clinics use isolated internal networks that communicate with the production servers at the primary site.

Figure 4-1 shows the geographical distribution of the company.

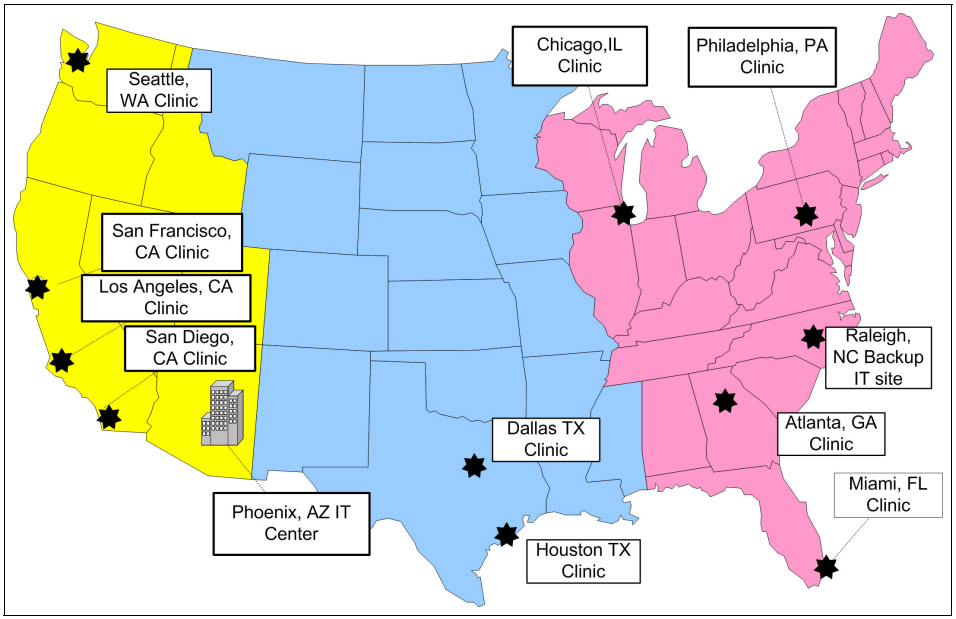

Figure 4-1 Geographical distribution of the cardio healthcare company

Clinics

The cardio healthcare company runs clinics in multiple US states. Each clinic operates its own network, with multiple zones, and communicates with the primary data center.

Primary data center

All customer-related information is stored on separate database entities that are clustered to fulfill high availability (HA) requirements. Most business critical applications are deployed in a highly available configuration.

Public web content is isolated on separate web servers and is not protected with Secure Sockets Layer (SSL). All existing network infrastructure components (such as firewalls, switches, and routers) are designed and implemented in an HA (redundancy) configuration.

Application servers and database servers (running DB2) are in separate network zones and are isolated from each other by firewalls.

The core applications used by external users (contracted medical staff that works remotely and business partners, such as pharmaceutical and research partners) are hosted in the Internet DMZ[1]:

- The Secure FTP server, which is used for information exchange with pharmaceutical companies
- A virtual private network (VPN) server that allows doctors to log on remotely to the company in an encrypted and authenticated way

In addition to these applications, internal users have access to additional applications:

- Applications that are locally deployed in each clinic on Windows Application Servers, including SharePoint applications used to report time sheets for teams that work in shifts (running in a Production Zone)
- SAP applications used in the billing process and for maintaining patient healthcare records (running in a Production Zone)

[1] DMZ is derived from demilitarized zone and stands for a subnetwork that exposes an organization's external services to the Internet.

Figure 4-2 shows the architecture diagram of the cardio healthcare company, which includes major communication lines between the separate network zones.

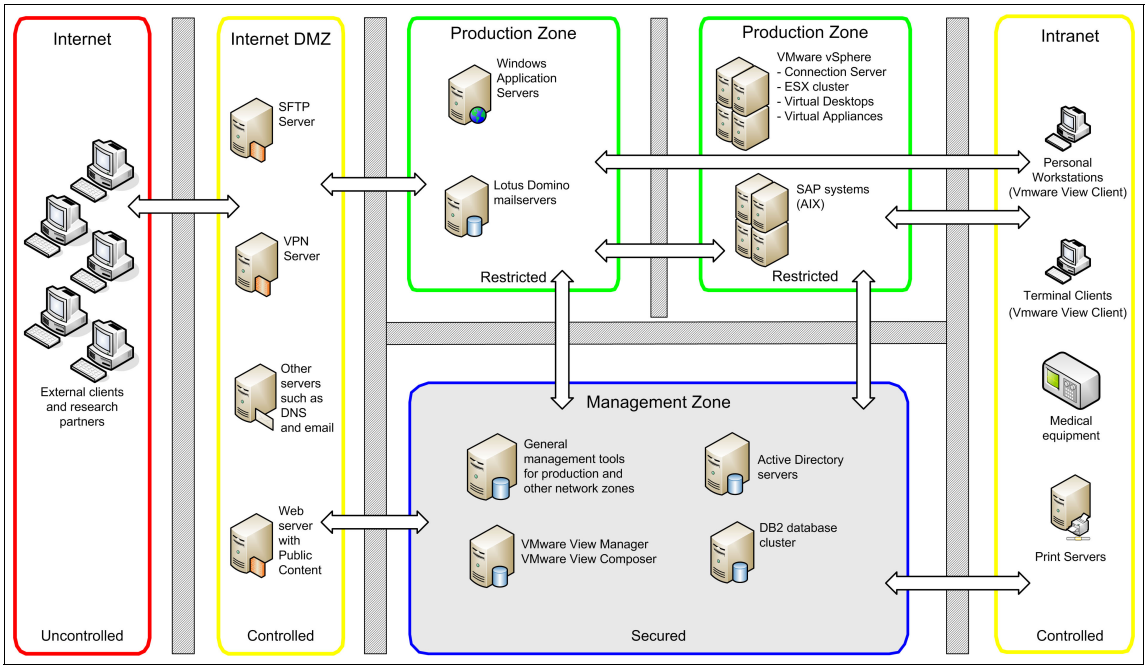

Figure 4-2 Current architecture overview of the cardio healthcare company

High availability capability: To simplify the diagram, we did not include any HA aspects of the current architecture.

Backup site

The backup site is not designed with high availability capabilities. The disaster recovery plan target is to recover the primary data center within one day. All system snapshots from the primary site are taken nightly and transferred to the backup site. In addition to this main purpose, the backup site is used as a test and integration environment for various application development and other IT projects.

Workstation solution design

The endpoint systems in the intranet network are primarily workstations that run Microsoft Windows XP or Windows 7. Many of those workstations are used by only one employee and are in closed office environments, physically shielded from the general public. These *personal workstations* offer a full office workspace

environment and have software installed that allows communication with various systems within the company.

A different type of workstation is in the semi-public areas of the various hospitals. These workstations are *shared* by the nursing (and other medical) staff on duty. Because these employees often need access to IT systems for brief periods of time only (for instance, to update patient progress reports), the company installed only a few systems per medical department.

Patients and visitors can gain physical access to these systems, because they are not kept behind closed doors. For this reason, Group Policy Objects (GPO) policies in an Active Directory environment are centrally deployed to lock down the functionality of these systems. The generic Windows account, which is automatically logged in on these machines when they are started, is configured with the principle of *least privilege*. The generic Windows account on these semi-public workstations does not allow software to be installed or run, except for a few explicitly listed applications. These *terminal clients* or *kiosks* can access only a few select resources. They allow Internet browsing and, more importantly, permit authenticated access to a *VMware virtual desktop infrastructure (VDI)* environment through a *VMware View Client*.

All employees of the company can log on to the secure virtual desktop environment from these shared semi-public workstations. Every employee is required to log on to their own virtual environment with their own account. They all have a persistent virtual desktop image linked to their unique account. From within this secure virtual desktop, they can access the necessary SAP systems and check their email.

Persistent images: A *persistent* virtual desktop image is linked to every account rather than a *volatile* image. This solution design allows employees to continue to run their applications in their virtual environment even when they log out of the virtual session, saving them time at logon. The drawback of using persistent images rather than volatile images is a higher required amount of memory on the Virtual farm and a higher Windows license cost. The company accepts this drawback because of the higher level of user productivity and satisfaction.

Whenever the medical staff that shares the shared workstations needs to interact with confidential data (private patient data and financial data), they must use the virtual desktop environment. Direct interaction between the workstations in semi-public areas and confidential data is impossible.

Another reason why the company created the virtual desktop infrastructure was for surgeons that work at the hospital (sometimes as independent contractors). These surgeons insist on using their own notebooks and tablet devices. The

cardio healthcare company security policy insists that no patient data can be stored on these systems. These personally owned devices are not managed by the company and are not granted the same level of trust. For that reason, the company insists that these employees also connect to a virtual desktop from their notebooks. From within the virtual environment, they can access the necessary systems securely.

4.1.2 Security and usability issues within the current infrastructure

While the move to a virtual desktop model solved many security concerns, the cardio healthcare company still faces many issues.

This virtual desktop model enables the company to build and maintain a standardized and secure environment, irrespective of the hardware or operating system deployed on the many workstations and personally owned notebooks. The shared kiosk stations at the various medical departments in the hospitals and the personally owned notebooks used by some physicians are detached from the valuable data. Only the secure virtual desktops can interact with the company data.

However, many concerns remain regarding user authentication:

- The corporate security policy defined strict policy standards for the most important applications. The standards mandate complex passwords for logging in to the virtual desktop through a VMware view client, logging in to the various SAP systems and applications (HR and finance), and using the Lotus Notes mail client. As a result, the personal workstation users are unhappy with the many complex passwords they need to memorize, frequently change, and must enter every time that they start an application. The user community wants to standardize and reduce the number of credentials they need to type. They want to focus on their jobs, and they want to see the IT department as an enabler rather than a hindrance.
- Due to the complexity and number of existing passwords, the IT support desk reports that many incidents that are raised day-to-day are password-related. Users forget passwords. They lock themselves out of applications by entering incorrect credentials several times in a row. Some employees even struggle with creating a complex password for the various applications. The support teams want to safely enable the users to reset forgotten passwords themselves rather than continue to guess with the risk of getting locked out.
- The medical staff (often nurses) that uses the shared terminal clients complains that they must log on to their virtual environment by entering the user name and password every time that they want to add a brief note on a patient record. They often need access through a terminal for little more than a minute and consider it a burden to enter their password again each time.

They prefer a faster and more convenient way to use the systems for their type of work.

- The corporate security officer is worried about the possibility of users that bypass the corporate security standards. Users might choose to keep their virtual desktop sessions open at unattended terminal workstations to avoid having to log on every time that they want to update a record after visiting a patient. Worse even, other staff members might choose to use the lingering virtual desktop sessions of other users to change patient records. This lack of trust in *who made which change* (person A using the account of person B) can cause the cardio healthcare company to fail its recurring compliancy audit. The company needs audit reporting, which can clearly link individuals to the actions that are logged.

4.2 Business vision

This section highlights the future direction to which the cardio healthcare company plans to move its business development in the next five years:

- Expand business to the European Union (EU) by opening a clinic in Munich, Germany.

  By collaborating with several pharmaceutical companies on research in the EU, the cardio healthcare company wants to expand its business related to heart diseases by opening a clinic in Europe. The project is scheduled to begin in two years, and the opening is planned in the next four years.

- Reduce costs by reusing the solutions and using the lessons learned from the current IT infrastructure.

  The cardio healthcare company wants to reuse its architectural and implementation approach wherever possible. While copying the general infrastructure design, it tries to improve challenges found during operations and remediate them at an early phase.

- Respond to changing business needs and technology directions that can help improve the user experience by adopting new technologies.

  The "*bring your own device"* (BYOD) trend in IT is noticed at the cardio healthcare company. An increased demand is registered from employees that request the use of personally owned notebooks, tablets, and smartphones.

  The myriad of changes requires a greater flexibility in IT technology. However, new technologies and means of communication open the possibility for new weaknesses.

- Manage the budget by avoiding penalties due to non-compliance with major regulations, such as the Health Insurance Portability and Accountability Act of 1996 (HIPAA).

  In the healthcare industry, as in many others, non-compliance with regulations and standards can lead to significant financial fines and other types of penalties. The cardio healthcare company is successful in managing compliance with major regulations and is looking to maintain this good practice while expanding the business.

- Protect the company image and reputation by avoiding patient information leakage, preventing security attacks, and practicing security due care and due diligence.

  Any unauthorized change to or leakage of any type of patient information (health, financial, or personal), can lead to a loss of trust and damage to the reputation of the cardio healthcare company. In addition to losing money on penalties, it leads to a loss of customers and missed revenue opportunities.

4.3 Business requirements

Based on the visionary aspects highlighted in 4.2, “Business vision” on page 96, and the information in 4.1.2, “Security and usability issues within the current infrastructure” on page 95, the cardio healthcare company wants to achieve the following short-term business goals:

- Improve the quality and availability of patient care and satisfaction by delivering an excellent, individualized healthcare experience.
- Increase the protection of all patient-related information, and address the diverse security risks that are driven by compliance requirements, emerging technologies, and data explosion.
- Facilitate the management and demonstration of the overall compliance posture with data privacy laws and industry regulations, such as HIPAA and PCI-DSS (Payment Card Industry Data Security Standard).

Overall, the cardio healthcare company wants mature security solutions that can prevent information leaks, and ensure trustworthy authentication and individual traceability and accountability of all actions that affect the patients.

By addressing these pressing business requirements, the cardio healthcare company is trying to achieve the following goals:

- Continue to manage an acceptable balance between preventing security risks and adversely affecting the business. User satisfaction levels are an important metric for IT projects at the cardio healthcare company.

- Constantly look for new and innovative solutions in all areas of the business, and always take security aspects into account.
- Be more proactive in security measures.
- Raise security awareness throughout the company by practicing corporate security education. Informed users are more likely to accept and support the security standards that are enforced.

4.3.1 IBM Security Framework mapping to business requirements

Based on the following information, the IT management team from the cardio healthcare company can articulate its needs better to design a solution:

- The IBM Security Framework definitions for business-driven security, which are introduced in 2.1.1, “IBM Security Blueprint perspective” on page 13
- The business requirements of the cardio healthcare company, which are outlined in 4.3, “Business requirements” on page 97
- The current organizational infrastructure, which is explained in 4.2, “Business vision” on page 96

Through this discussion, the assigned IT architects can derive the functional requirements by using the underlying IBM Security Blueprint:

- People and Identity

  The cardio healthcare company wants to uses mature identity management and authentication processes and tools that help lower the costs related to this domain. The company wants to maintain its security standards without putting undue stress on the user community. The single sign-on solution that we design in the following chapters for the cardio healthcare company mainly relate to this domain.

- Data and Information

  The cardio healthcare company uses a granular information asset classification scheme paired with a least privilege principle. Access to the SAP and database servers is strictly real-time monitored and enforced.

- Application and Process

  Application development focuses on the *secure by design* principle. The cardio healthcare company follows a rigorous release management process with a granular promotion-to-production path that specifies security testing criteria. This approach helps with practicing security during the application development phase and helps to discover any application vulnerabilities.

The processes of the cardio healthcare company achieved a high level of automation and embrace security controls, such as the separation of duties and creation of auditable records.

- Network, Server, and Endpoint

  The cardio healthcare company deployed network segregation. Systems are assigned to a security zone according to the sensitivity of the system and the data on it. The various security zones are separated by firewalls and intrusion prevention systems. Remote users must authenticate to a VPN server before entering the network.

- Physical Infrastructure

  Physical Infrastructure controls are also embraced in the security program of the cardio healthcare company. Respective controls for physical access controls to facilities and systems are also present in all locations. All systems that can access data of value (including the personal data of patients) are either in closed office environments or in a secure virtual environment.

- Governance, Risk Management, and Compliance (GRC)

  The cardio healthcare company practices strong compliance enforcement by managing a security controls framework and a strict audit and security awareness program. Adequate audit-reporting is required to satisfy the auditors that assess the company's compliance with important regulations for the company operations. Adequate audit-reporting must be available in the solution that we design in 4.5, "Design approach" on page 103.

We now take the next step to understand the functional requirements and map them to the IBM Security Blueprint, followed by a high-level look at the implementation approach.

4.4 Functional requirements

To properly address the new business requirements, the cardio healthcare company must enhance its authentication solution infrastructure. The cardio healthcare company defines the following high-level functional requirements:

- To better manage its compliance posture with data privacy laws and industry regulations, the company must employ a cost-effective centralized management solution for user-centric event logging. It must be able to use the proposed new security solution to generate the necessary audit reporting that can demonstrate its compliant state. The company must be able to map all actions related to patient data to individual employees by ensuring that all actions are linked to people who are who they claim to be.

- To improve the quality and availability of patient care and satisfaction by delivering an excellent, individualized healthcare experience, and to increase caregiver productivity and reduce administrative costs, the company must address delays in the IT authentication processes. The cardio healthcare company wants to build a solution that allows the company to centrally provision users for various systems in accordance with policy-driven security standards for secure authentication. The company also want users to be able to reset their own passwords without having to call a costly support desk. The company wants to reduce the number of passwords that users need to create and memorize, without compromising security.
- To alleviate the stress put onto the staff of having to maintain, memorize, and enter many credentials, the company wants to build a solution to allow alternative secure authentication factors other than passwords. The company wants a secure authentication system that allows faster user logon and faster user switching. This solution needs to fully support the Virtual Desktop environment that the company rolled out throughout its hospitals. The company also insists on a method to ensure that unattended terminal clients are automatically disconnected.

The cardio healthcare company already uses some solutions to securely facilitate user authentication for roaming users. Nurses that share workstations can continue to run their applications through the persistent virtual desktop images, but they still need to enter a password every time that they want to use a terminal station.

4.4.1 IBM Security Blueprint mapping to functional requirements

We now understand the functional requirements for the additional security measures that the cardio healthcare company needs to implement. Yet, we still must determine which specific solutions can potentially fulfill the functional requirements. By using the IBM Security Blueprint, we can better explain and map the functional requirements into specific blueprint areas, identifying the appropriate solutions to implement within the IT environment of the company.

Figure 4-3 on page 101 shows the mapping of the functional requirements to the IBM Security Blueprint.

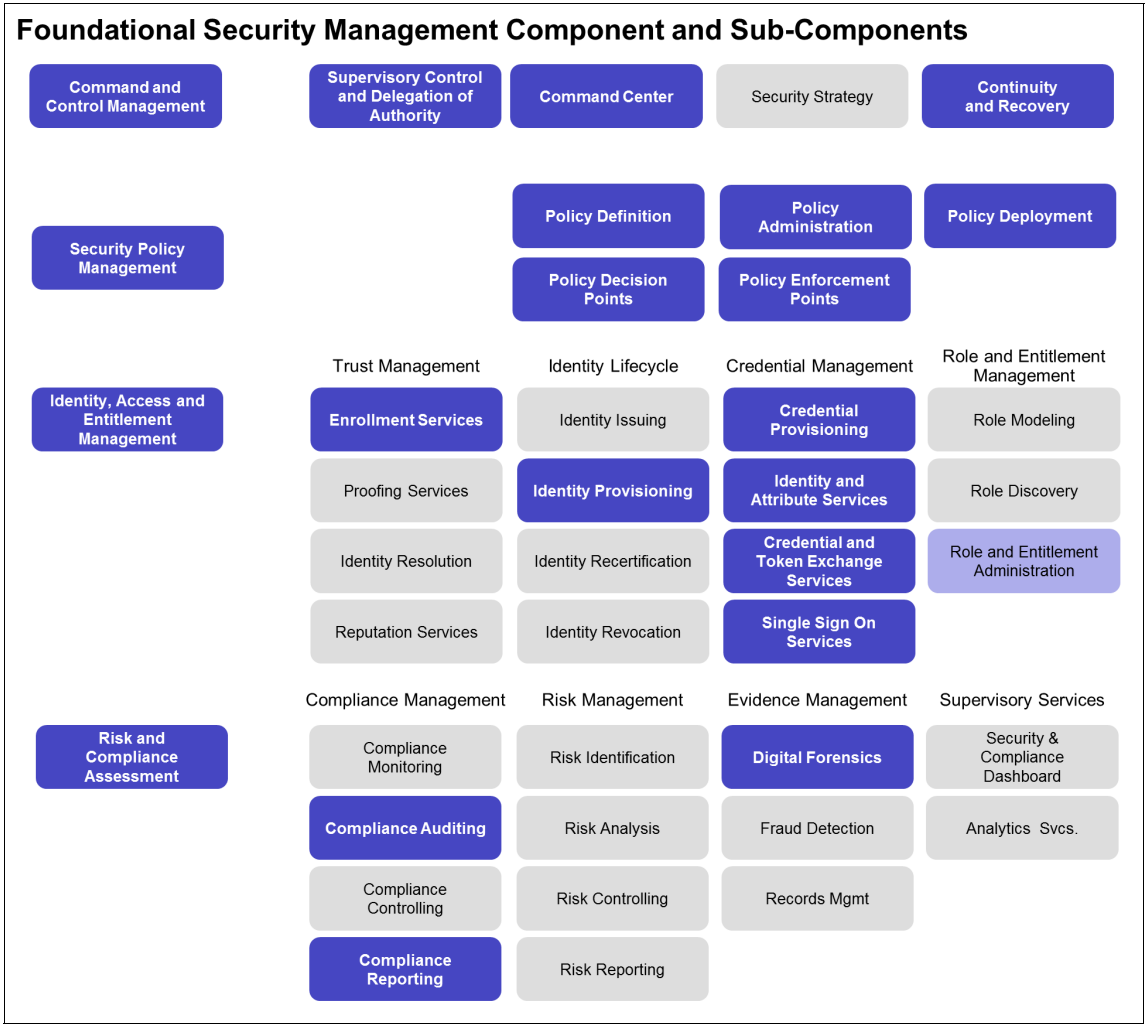

Figure 4-3 IBM Security Blueprint: Foundational components for functional requirements

Let us look at the functional requirements derived from the IBM Security Blueprint and map them to each of the required Foundational Security Management components and subcomponents:

- Define the various roles within the solution that must be managed. Create distinct permissions for the roles of users, administrators, and help desk operators (Supervisory Control and Delegation of Authority).
- Continue good management practices by adding a centralized management platform (IBM Command Center®) for managing user authentication credential sets, logging, and reporting.

- Create a true enterprise solution that allows the creation of highly available centralized systems and also allows for prompt disaster recovery in the backup data center (Continuity and Recovery).
- Define and enforce the human-readable corporate security policy and authentication standards in machine-readable logic. Ensure that this policy is enforced on all relevant systems and applications, even those applications that natively do not support this policy without altering the code base (Security Policy Management).
- Allow users to register biometric details or radio frequency identification (RFID) badges to be linked to their account (Trust Management).
- Integrate with user provisioning systems for new users (Identity Lifecycle).
- Make sure that user identities can automatically be assigned correctly formatted passwords and be linked to other authentication factors, such as access badges that are issued. Allow integration with existing Directory services (Credential Management).
- Support custom and pre-defined reporting on log data on authenticated users for audit purposes. Allow for on demand and scheduled exporting of reports (Compliance Management).
- Generate and securely store trustworthy and individually accountable log data (Evidence Management).

Although business and functional requirements are the main parts of the security design objectives, we also must consider other non-functional requirements and constraints. These non-functional requirements and constraints might include objectives that are necessary to meet general business requirements, service level agreements (SLAs), or practical constraints about constructing security subsystems.

The architectural team analyzed the non-functional requirements and identified the following key non-functional requirements and constraints:

- Allow for 99.999% (“five nines”) availability (access to the system). HA is not included in the initial design, but moving to HA later must be possible without a new setup phase.

Disconnected operations: AccessAgent deployed to workstations can fully operate disconnected from IBM Security Access Manager for Enterprise Single Sign-On server components. Only actions, such as central policy updates, do not occur when the server components are unavailable. Therefore, many organizations choose not to build an HA setup and rely on disaster recovery (DR) procedures only.

- Provide a scalable solution that can be replicated to any future expansion of the company, including worldwide hospital onboarding.

Product mapping: Because this book focuses on the security architecture of the People and Identity domain, it does not look in detail at these non-functional requirements. However, all IBM products mapped to this IBM Security Framework component can satisfy the non-functional requirements and constraints we mention.

The following sections show how to further use the IBM Security Framework and IBM Security Blueprint in both the design and implementation of new security solutions.

4.5 Design approach

The IT architect for the cardio healthcare company determined the areas of the IBM Security Blueprint that the new solutions must fulfill to adequately address all of the company requirements. The administrator for the company can now map the technical requirements to the Security Services and Infrastructure components of the IBM Security Blueprint. The purpose of this exercise is to help determine which security solutions best satisfy all of the requirements, whether they are business, functional, non-functional, or technical.

Figure 4-4 shows how the mapping was done for the cardio healthcare company by using the functional requirements and existing architecture.

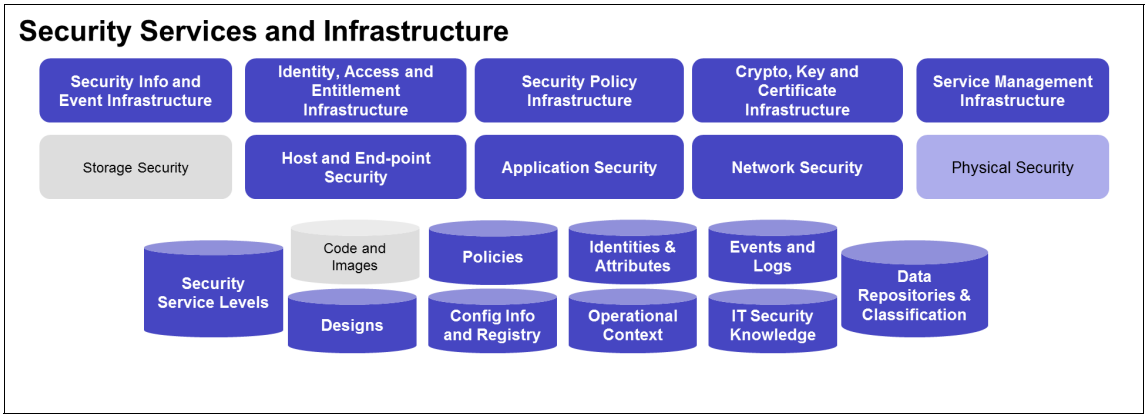

Figure 4-4 IBM Security Blueprint: Technical components for design

As part of the design, the company can produce an implementation plan for the deployment that involves the following steps:

1. Prioritize the requirements.
2. Map the requirements to IBM product 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.

Therefore, the company can now focus on the technical components of the IBM Security Blueprint and how they can be mapped into technical and operational requirements.

Based on the mapping, the company now knows that the solutions must provide the following Security Services and Infrastructure components:

- Integrate with the existing Identity directory services deployed.
- Support the existing endpoint technologies (operating systems and applications).
- Provide long-term tamperproof storage of event data that logs access attempts and whether they were granted.
- Centralize management of technical-level security polices with a single location to design, update, change, and roll back the policies.
- Ensure secure communication between key components by using a certificate infrastructure or some type of public/private key infrastructure.
- Deliver a tamperproof environment on endpoints and servers to safeguard authentication credentials both in memory and when not in use.

> **Physical and storage security:** Physical security and storage security are also important parts of the overall solution. However, they are treated as separate projects that are not in the scope of this book.

- Provide an efficient mechanism for maintaining a record of the actual configuration.
- Provide a multitier architecture that can inherit the fault-tolerance of the existing infrastructure (including databases and directory servers).
- Establish educated resources that possess the appropriate security knowledge and analytical skills and that have support from the vendor.

After completing the mapping, the company can use the output to determine which solutions are needed and ultimately produce an implementation plan for

the selected solutions. To accomplish this task, the company must map all of the stated requirements into product features by using the IBM Security Blueprint diagrams.

For more detailed information about using the IBM Security Blueprint, see *Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, REDP-4528.

4.6 Implementation approach

You now understand all of the requirements and how they map into the IBM Security Framework and IBM Security Blueprint. The IT architect for the cardio healthcare company can apply this knowledge to select the appropriate solutions to satisfy all of the requirements.

The remainder of this book focuses on the implementation plan steps to set up the IBM Security Access Manager for Enterprise Single Sign-On systems. This approach results in distinct deployment phases:

1. Set up the central IMS Server and configure it for use. Create and set up the required AccessProfiles, as discussed in Chapter 5, “Base installation and configuration” on page 107.
2. Implement the password reset service as discussed in Chapter 6, “Password self-services implementation” on page 199.
3. Create the necessary *machine policy templates* and *user policy template* that enable the user-driven onboarding of multiple factor authentication mechanisms, as discussed in Chapter 7, “Strong authentication using RFID” on page 227.
4. Enable the use of single sign-on in the virtual desktop environment (VMware VDI), as discussed in Chapter 8, “Roaming desktop implementation” on page 259.
5. Perform daily operational tasks, such as generating audit reports, maintaining the database, and generating disaster recovery backups, as discussed in Chapter 9, “Implementing operational requirements” on page 271.

4.7 Conclusion

This chapter combined several IBM Security Systems products to help an organization to fulfill the requirements for Identity, Access, and Entitlement Management. It showed how the IBM Security Framework and the IBM Security

Blueprint can provide a structure to derive the IT functional and technical requirements from the business vision, goals, and requirements.

First, this chapter introduced the cardio healthcare company. It began with a company profile, its current IT infrastructure, and its issues with people and identity. Then, the chapter explained the business requirements and the associated functional requirements, including refining them to a more detailed technical level.

Next, this chapter described the design approach that the cardio healthcare company took for its solution, following the IBM Security Framework and the Single Sign-On Solution Pattern of the IBM Security Blueprint. When applied to the unique IT environment of the company, this process of analysis and design helped the cardio healthcare company define an implementation plan.

5

Base installation and configuration

In this chapter, we describe the technical implementation of the IBM Security Access Manager for Enterprise Single Sign-On base environment. First, we explain how to install the necessary components. Then, we describe how to deploy the enterprise single sign-on setup. Finally, we explain how to manage and maintain the deployed solution.

5.1 Design considerations

In this section, we focus on the concepts of a base-level implementation of IBM Security Access Manager for Enterprise Single Sign-On, and the components you must be aware of when designing the deployment architecture. Consider the existing infrastructure at the cardio healthcare company, described in 4.1.1, “Current IT infrastructure” on page 91.

The following components are required for a base-level implementation of IBM Security Access Manager for Enterprise Single Sign-On:

- Central user repository/directory

  The central user repository can be one of several supported repositories, including Active Directory, Novell, and generic Lightweight Directory Access Protocol (LDAP). The central user repository must be in place before installing any IBM Security Access Manager for Enterprise Single Sign-On components. In the cardio healthcare company environment, the central user repository is Active Directory.

- IMS Server

  The IMS Server is a Java-based application that runs on its own instance of IBM WebSphere Application Server. It can be a software installation on a Windows server platform or integrated in a packaged virtual appliance.

- IMS database

  The IMS database stores all of the IBM Security Access Manager for Enterprise Single Sign-On configuration, policy, and user data. This database can be created on an existing database server, or it can be installed on the same system with the IMS Server. Supported databases include IBM DB2, Microsoft SQL, and Oracle. In the cardio healthcare company environment, an existing DB2 database is used as the database.

- AccessAgent

  An AccessAgent is installed on each client system, Windows Terminal Server, VMware Virtual Desktop, and Citrix XenApp server that is to be managed by IBM Security Access Manager for Enterprise Single Sign-On.

- AccessStudio

  AccessStudio is an administrative tool used to create AccessProfiles. It needs to be installed on one workstation only, normally on the workstation of one or more IMS Server administrators. Because AccessStudio requires AccessAgent, install AccessAgent on the same workstation before you install AccessStudio.

5.1.1 System requirements

The IBM Security Access Manager for Enterprise Single Sign-On base components can be integrated onto existing servers if the servers have sufficient resources.

For specific software and system requirements, see the *IBM Security Access Manager for Enterprise Single Sign-On Planning and Deployment Guide Version 8.2,* SC23-9952-03.

5.1.2 Deployment architecture

The deployment architecture for an IBM Security Access Manager for Enterprise Single Sign-On-based installation consists of a client-side application (AccessAgent) that communicates with a central server-side application (IMS Server). Deployments can become more complex with the integration of optional advanced components, such as identity management software and external data sources. Even so, the client-server model remains the same for the core IBM Security Access Manager for Enterprise Single Sign-On components.

Client-side components

IBM Security Access Manager for Enterprise Single Sign-On consists of two client-side applications: *AccessAgent* and *AccessStudio*.

AccessAgent is installed on user workstations and Microsoft Terminal, VMware Virtual Desktop, or Citrix XenApp servers. Its main function is the recognition and interception of user authentication and change password dialogs. It acts on these dialogs for authentication and password change automatically depending on how policies are configured. AccessAgent consists of several underlying components that also perform these tasks:

- Data synchronization with the IMS Server for updating policies and profiles, and retrieving user Wallets
- Secure storage of credentials in the Wallet on the local workstation

The underlying components and their architecture are discussed in detail in 2.2.1, “AccessAgent” on page 17.

AccessStudio is a tool that allows administrators to configure or create *AccessProfiles*. These profiles facilitate the automatic logon, logoff, and password change for applications that require authentication. AccessStudio must be installed on only one administrative workstation.

Server-side components

The server-side components consist of the *IMS Server* and the *IMS database*.

The IMS Server is the central point of administration for user identities, AccessProfiles, authentication policies, and authentication factors. The administration is through a web interface called *AccessAdmin* where administrators can create and modify policies, and manage users.

The IMS database stores all IBM Security Access Manager for Enterprise Single Sign-On configuration and user objects, such as policy templates, user credentials, authentication services, and AccessProfiles. How user credentials are securely stored on the database and in local user Wallets is described in detail in “Securing Wallets” on page 62.

AccessAgent synchronizes with the IMS Server on a regular interval to retrieve policy updates. The predefined synchronization interval is 30 minutes. It can be changed to 5 minutes, for example. However, this change is not suggested and unnecessary, especially because the additional network traffic must be considered.

Target applications

The target applications can typically be grouped into the following categories:

- Windows client/server

  Typical application with a client component that is locally installed on the user workstation. The client component requests user authentication and communicates with an application component that runs on a back-end server, for example, Lotus Notes.

- Java based

  The authentication dialog for this type of application was developed in Java and is sent to and executed on the user workstation at application start time.

- Web-based

  Applications that run on a web server that requests users to authenticate from a web browser.

- Terminal emulators

  Terminal emulators are installed and executed locally on client workstations and are configured to communicate with back-end applications that emulate a specific terminal type. Examples are 3270 terminal emulators to access host-based applications and Telnet to access UNIX systems.

In the first phase, the cardio healthcare company decided to implement support for the following applications:

- Lotus Notes client
- SAP client
- VMware View Client

5.2 Installing and configuring base components

In this section, we discuss the installation and initial configuration of IBM Security Access Manager for Enterprise Single Sign-On. The following steps are involved in installing and configuring the base components:

1. Installing and configuring a database for use by the IMS Server, and adding the necessary administrative users for the database and initial IMS Server configuration
2. Installing and initially configuring the IMS Server
3. Installing and configuring AccessStudio
4. Using AccessStudio to create AccessProfiles
5. Configuring policies to enable single sign-on capabilities for the cardio healthcare company employees
6. Installing and configuring AccessAgent on a client workstation
7. Users creating their IBM Security Access Manager for Enterprise Single Sign-On accounts and initiating the single sign-on process for the configured applications

5.2.1 Creating administrative users

To prepare for our base component installation and configuration, two administrative users must be defined in the central user repository:

- Database administrator

  The IMS Server performs all database operations on behalf of the user defined as the database administrator.

- Active Directory *lookup user*

  IBM Security Access Manager for Enterprise Single Sign-On uses the lookup user to retrieve user attributes from the Active Directory enterprise repository. The user defined as the lookup user must not be the primary user account for any employee, because password change or account lockout can cause problems with authentication for all users. A preferred practice is to create a

system account specifically to act as the lookup user. IBM Security Access Manager for Enterprise Single Sign-On uses the lookup user to search for users in the directory and to retrieve user attributes from the Active Directory enterprise repository.

Important: If the corporate security policy requires that the password of the lookup user must be changed at predefined intervals, the IMS administrator must be aware of this policy and set the new password in the IMS Server configuration.

5.2.2 Deploying the IMS Server Virtual Appliance

Because the cardio healthcare company invested in a scalable and redundant ESXi environment, the company chooses to install the IMS Server Virtual Appliance rather than manually install software on a newly provisioned Windows server. For platform requirements, see the *IBM Security Access Manager for Enterprise Single Sign-On Planning and Deployment Guide Version 8.2,* SC23-9952-03.

Follow these steps to deploy the IMS Server Virtual Appliance:

1. The VMware vSphere Client is used to log in to the ESXi server. Start the VMware vSphere Client and use the dialog shown in Figure 5-1 to log in. You need to specify an IP address or the name of the ESXi server in your environment, in this case, `192.168.25.136`. Click **Login**.

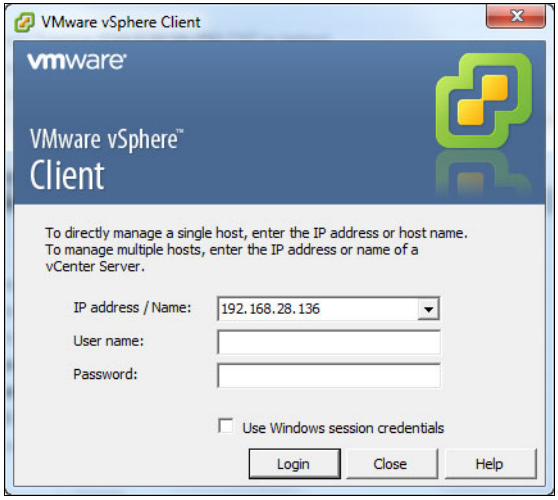

Figure 5-1 VMware vSphere Client login

2. After logging in to the ESXi server, click **File** → **Deploy OVF Template**, as shown in Figure 5-2.

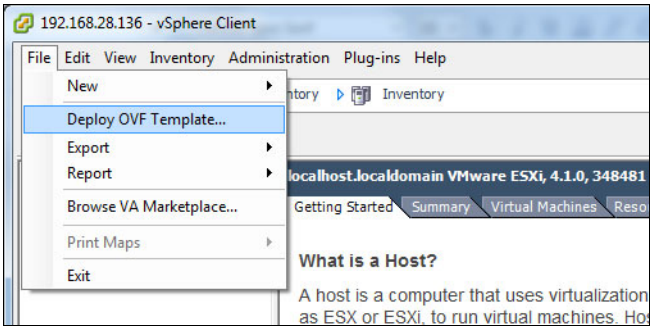

Figure 5-2 Deploy OVF Template

3. In this dialog, you need to select the extracted IBM Security Access Manager for Enterprise Single Sign-On Virtual Appliance. Click **Browse** and navigate to the appropriate file location to choose the ovf file. Then, click **Next**, as shown in Figure 5-3.

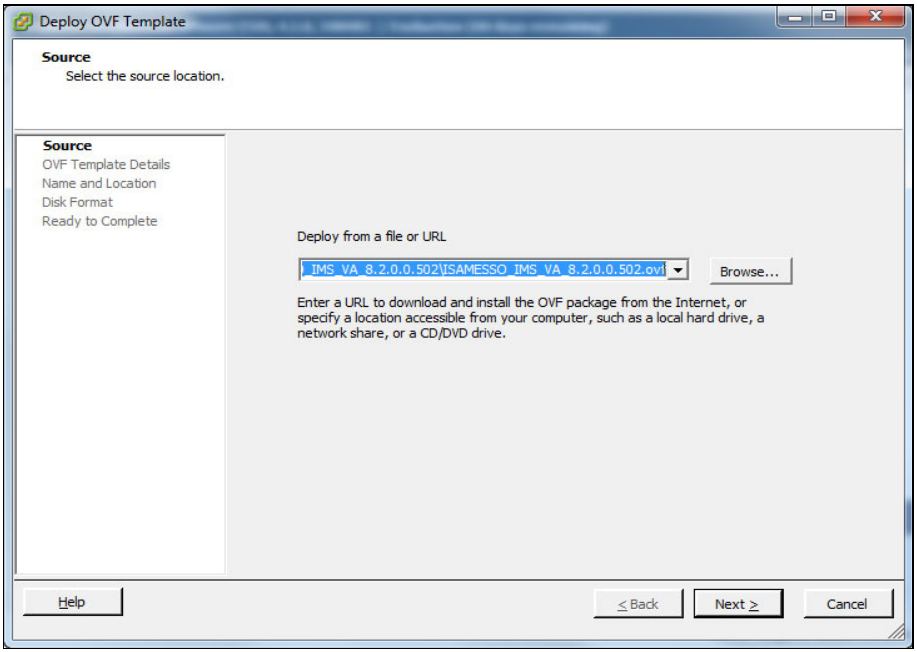

Figure 5-3 Deploy OVF Template

4. Click **Next** after verifying the details, as shown in Figure 5-4 on page 114.

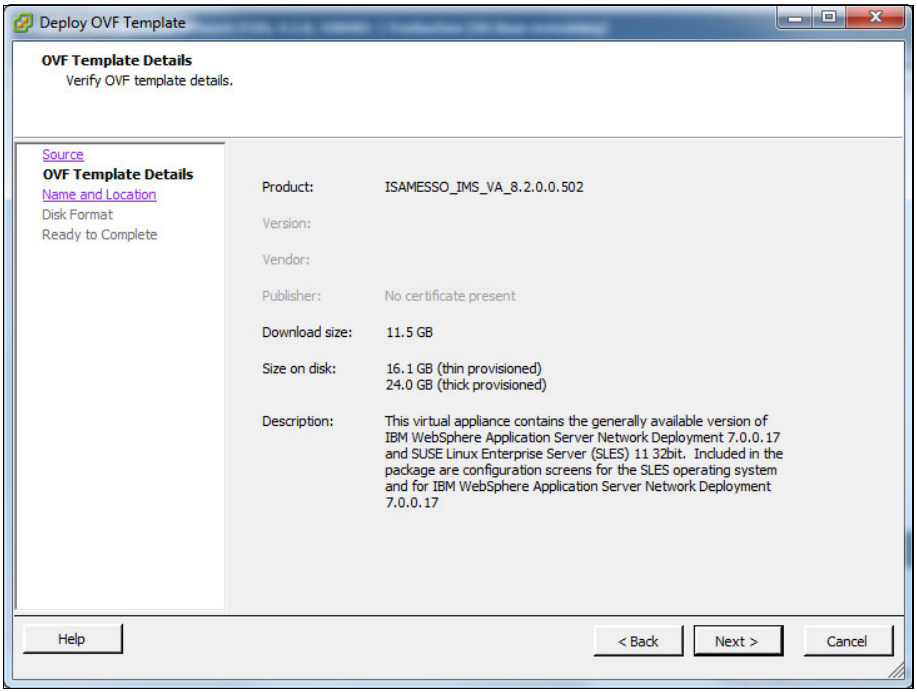

Figure 5-4 Verify OFV template details

5. Select a name for the Virtual Appliance and click **Next**. The cardio healthcare company keeps the default name `ISAMESSO_IMS_VA_8.2.0.0502` as shown in Figure 5-5 on page 115.

> **Naming:** Be sure to use a unique name for every virtual appliance.

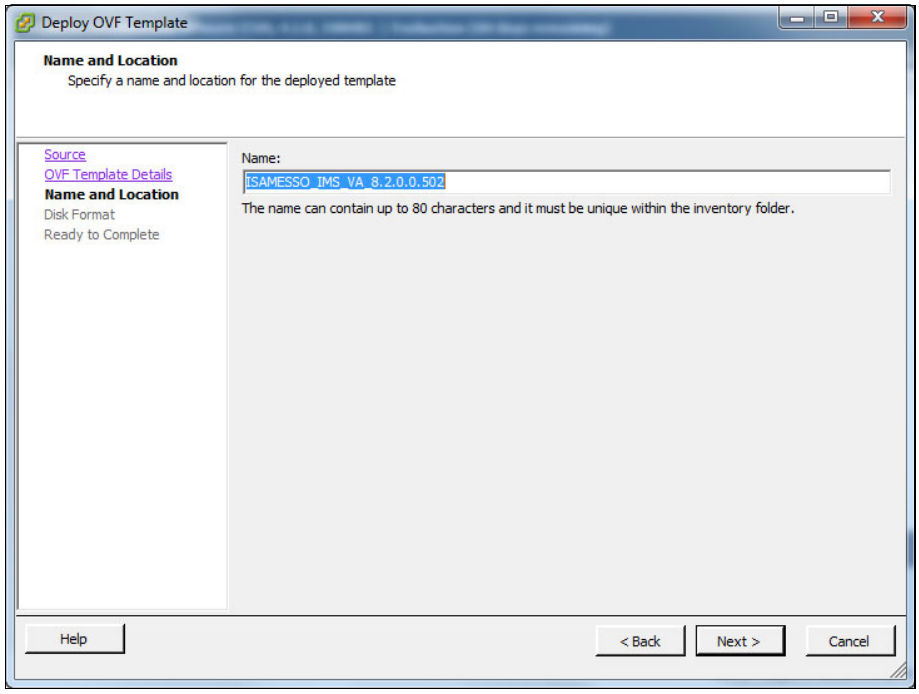

Figure 5-5 Select OVF template name

6. Select between a thin-provisioned format or a thick-provisioned format, and click **Next**, as shown in Figure 5-6 on page 116. Select **Thick provisioned format** to allocate all storage on the ESX server immediately, and click **Next**.

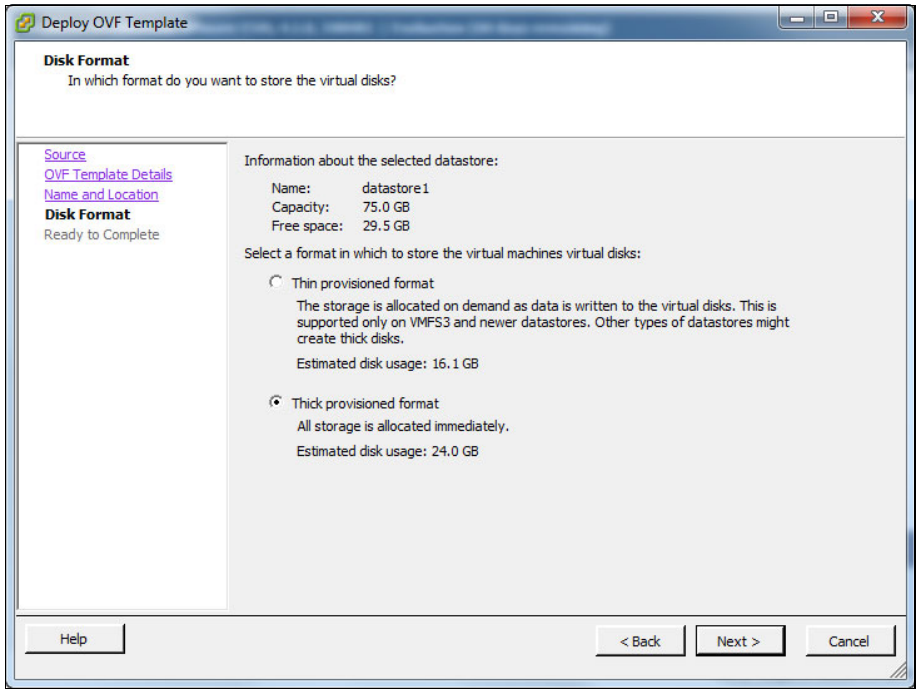

Figure 5-6 Select OVF template disk format

7. Verify the settings and click **Finish** (Figure 5-7 on page 117).

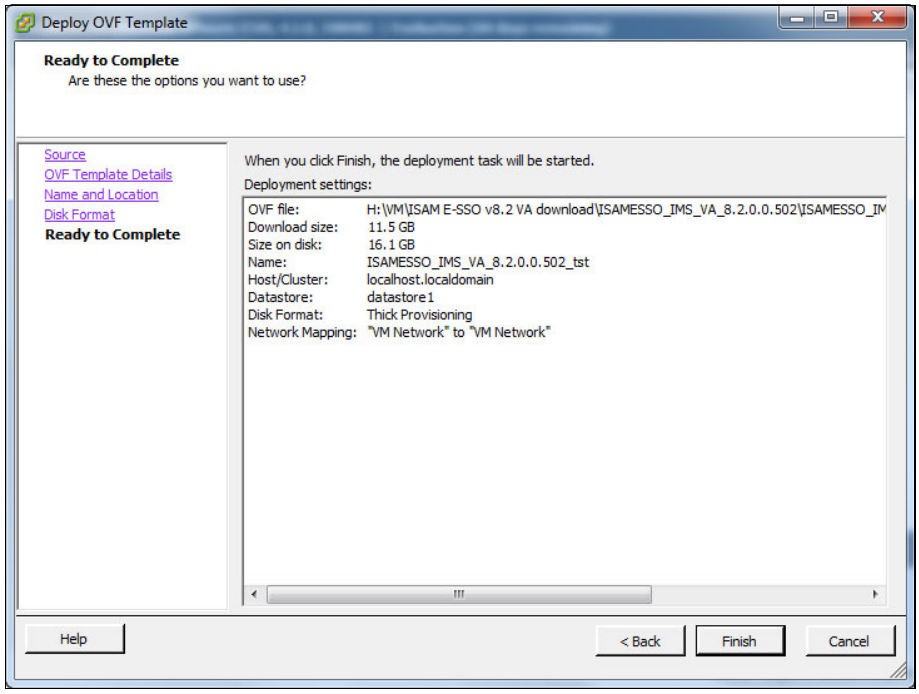

Figure 5-7 Verify settings and finish OVF template import

8. The import of the Virtual Appliance into VMware ESXi starts, as shown in Figure 5-8. This process can take a few minutes to complete.

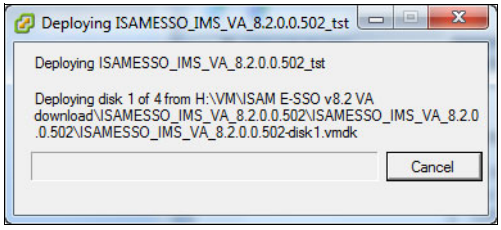

Figure 5-8 Deploying OVF template

5.2.3 Starting the Virtual Appliance

After a successful import, you can start and configure the virtual appliance:

1. Start the Virtual Appliance in the VMware vSphere Client. The Linux platform on which the Virtual Appliance is built starts, as shown in Figure 5-9 on page 118.

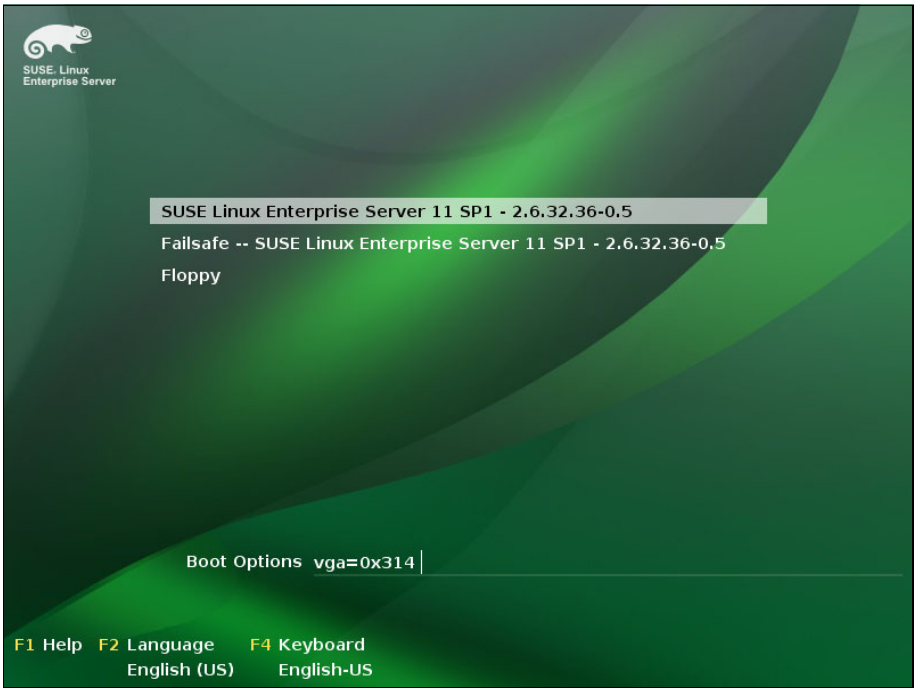

Figure 5-9 Starting the Virtual Appliance for the first time

2. Select the language, and click **Next** (Figure 5-10 on page 119). The cardio healthcare company keeps the default settings.

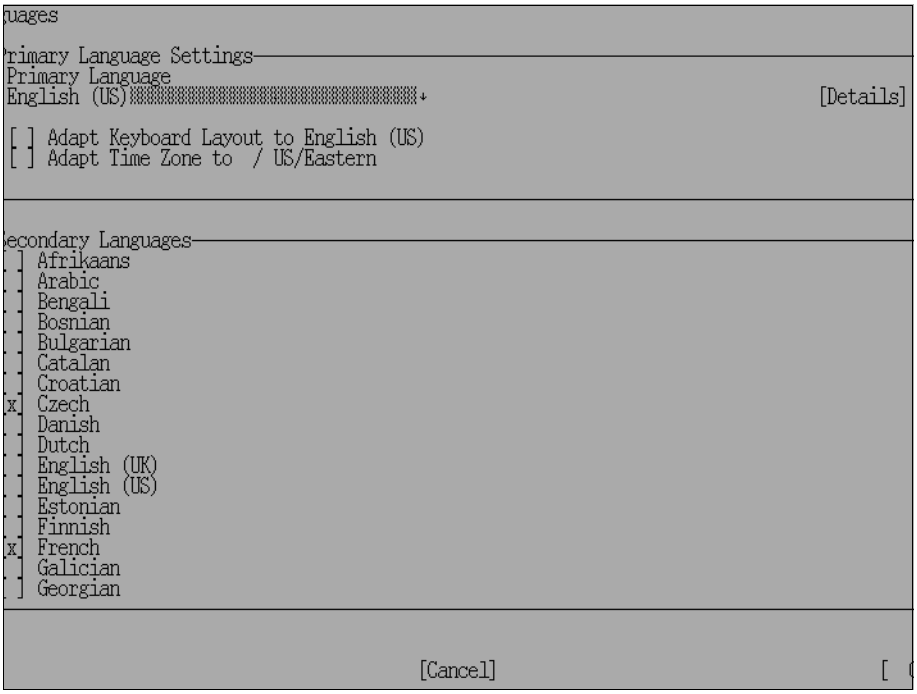

Figure 5-10 Select language settings

3. Accept the Linux Distribution Statement, and click **I understand** (Figure 5-11).

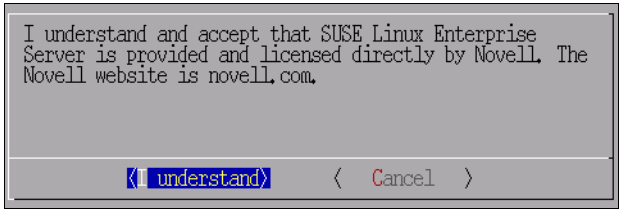

Figure 5-11 Accept Linux Distribution Statement

4. Read the software license agreement, and click **Next** (Figure 5-12 on page 120).

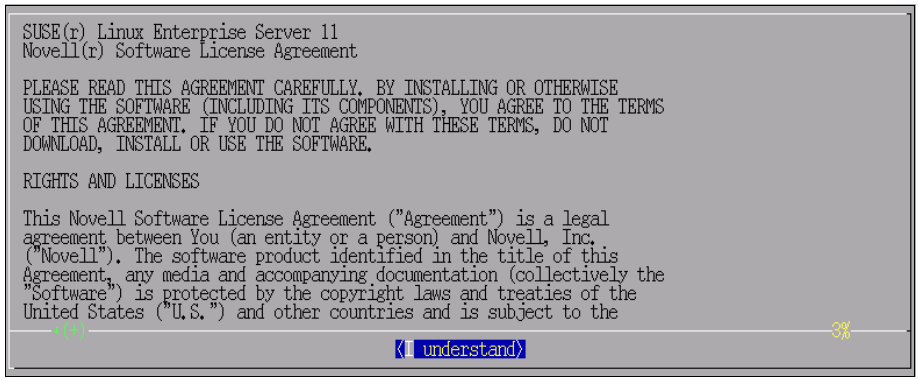

Figure 5-12 Software license agreement

5. Read and understand the VMware license agreement, and click **I understand** as shown in Figure 5-13.

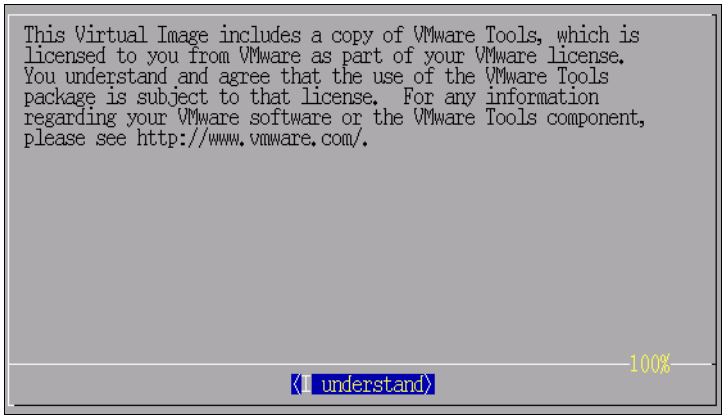

Figure 5-13 Read and understand the VMware license agreement

6. Read and accept the **International Program License Agreement**, and click **I understand** (Figure 5-14 on page 121).

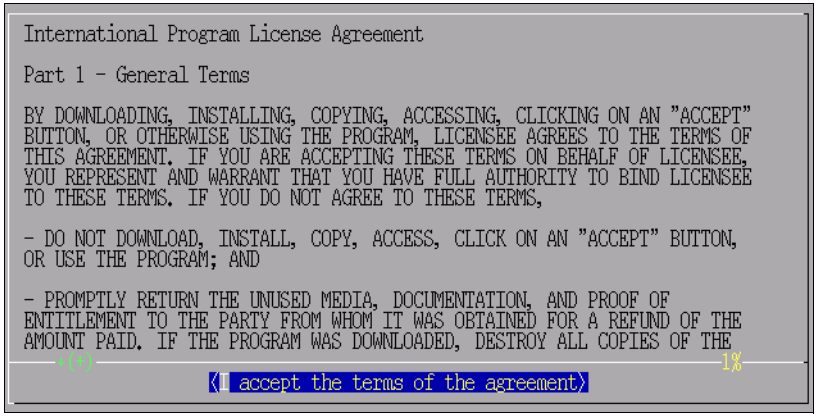

Figure 5-14 International Program License Agreement

7. Click **accept** to accept the license agreements for Novell SLES, VMware Tools, and WebSphere Application Server Hypervisor Edition, as shown in Figure 5-15.

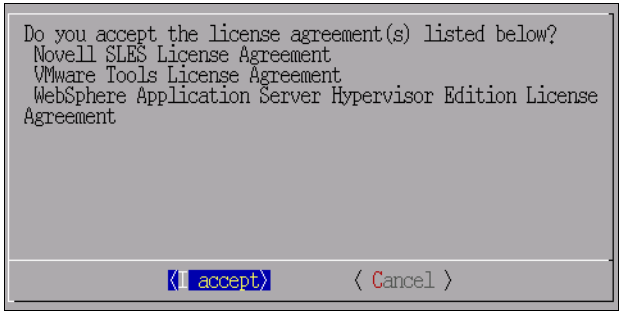

Figure 5-15 License agreements for Novell, VMware, and WebSphere

8. Choose the network protocol. The cardio healthcare company selects static network protocol for static IP addresses (Figure 5-16).

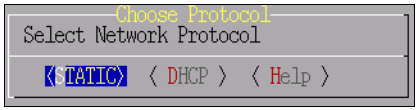

Figure 5-16 Select static network protocol

Note: During the deployment and activation of the virtual appliance, if you do not accept any of the license agreements, the virtual appliance shuts down. If you restart the virtual appliance, the language settings and license agreements previously configured and accepted are not displayed.

9. Enter the network parameters, and click **OK**. The cardio healthcare organization uses the parameters that are shown in Figure 5-17:

IP Address	192.168.28.135
Net Mask	255.255.255.0
Gateway Address	192.168.28.1
DNS	192.168.28.130
Host Name	imsva
Domain	cardio.example.com

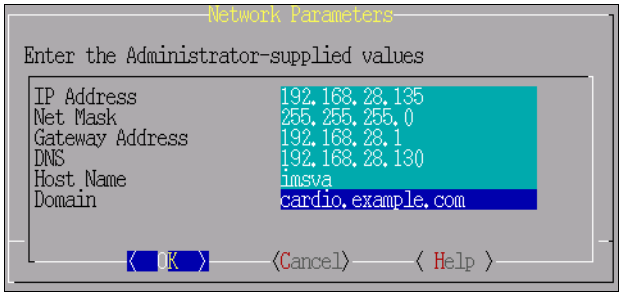

Figure 5-17 Enter the network parameters

10. Verify the network parameters, and click **Yes** (Figure 5-18).

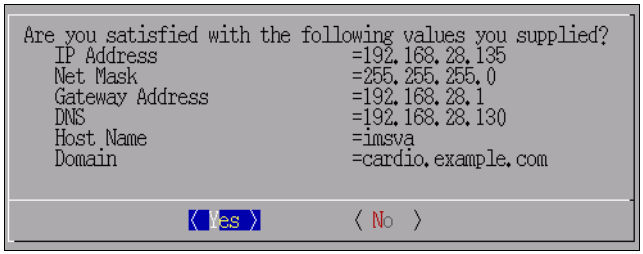

Figure 5-18 Verify network parameters

11. Enter a root password, and click **OK**, as shown in Figure 5-19 on page 123. In the next window, confirm the password.

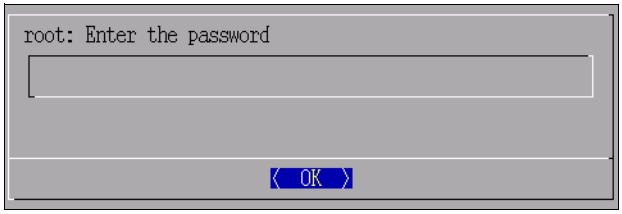

Figure 5-19 Set a root password

12. Define a default virtual image user ID. The company uses the predefined value `virtuser` (Figure 5-20). Click **OK**.

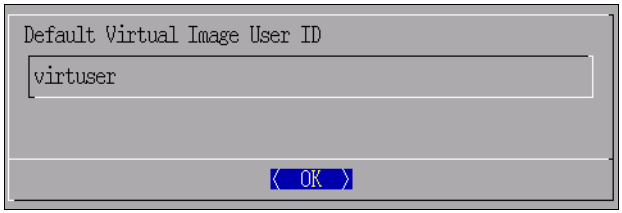

Figure 5-20 Define a virtual image user ID

13. Define a password for virtuser, and click **OK** (Figure 5-21). In the next window, confirm the password.

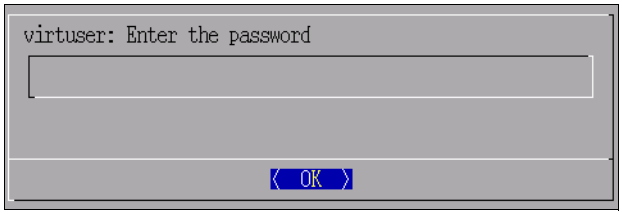

Figure 5-21 Define a password for the virtual image user ID

14. Next, define the timezone settings and click **F10** to continue. We keep the default settings (Figure 5-22 on page 124).

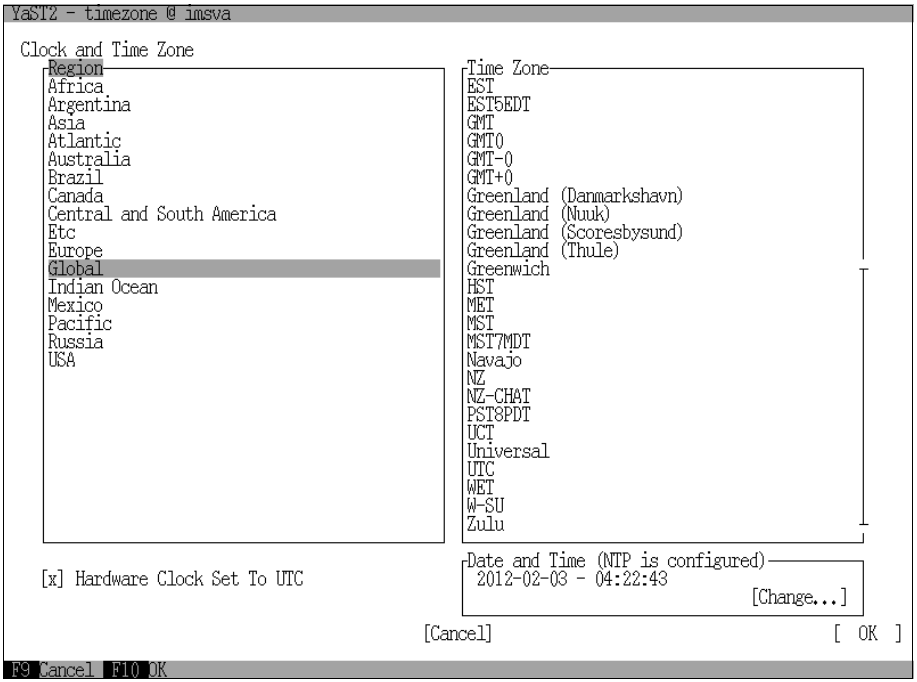

Figure 5-22 Define time zone settings

15. Choose between the IBM Security Access Manager for Enterprise Single Sign-On Standard license type and the Suite license type. The cardio healthcare organization purchased the Suite license type, which includes features, such as password reset. Check **Suite** and click **Next**, as shown in Figure 5-23.

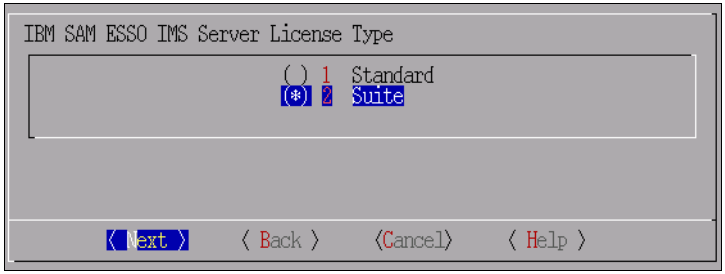

Figure 5-23 Select the IMS Server license type

16. Accept the terms of the license agreement for the IBM Security Access Manager for Enterprise Single Sign-On IMS Server License Agreement (Figure 5-24 on page 125).

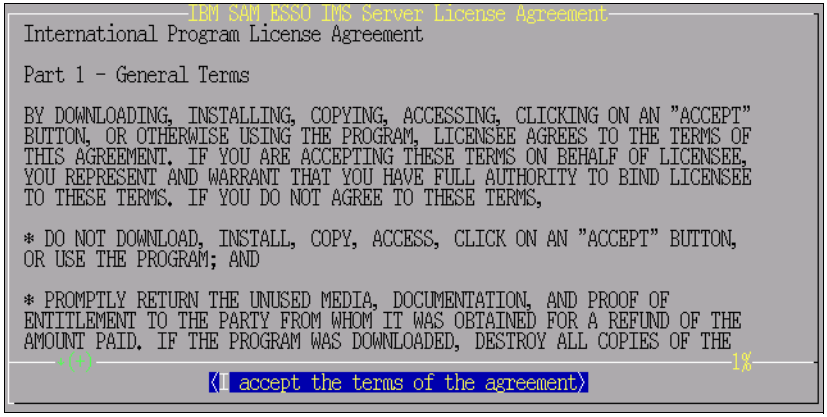

Figure 5-24 IBM SAM ESSO IMS Server License Agreement

17. Press the Space bar to install Tivoli Common Reporting (TCR) for IMS, as shown in Figure 5-25, and click **Next**.

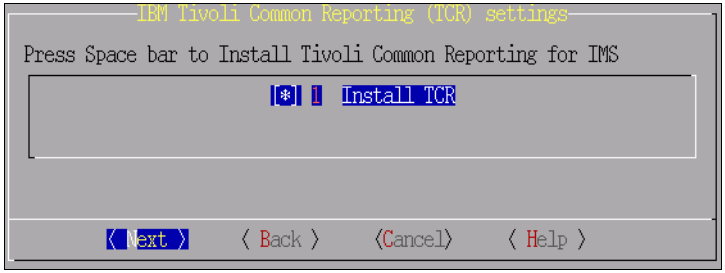

Figure 5-25 Install Tivoli Common Reporting (TCR) for IMS

18. The cardio healthcare organization decided to use Tivoli Common Reporting. Click **Yes** to confirm the installation (Figure 5-26). Tivoli Common Reporting is installed as a separate product. However, it is included in the IBM Security Access Manager for Enterprise Single Sign-On virtual appliance.

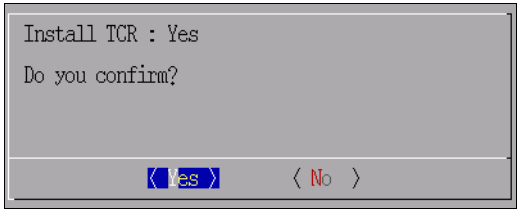

Figure 5-26 Confirm Tivoli Common Reporting installation

19. Finally, accept the Tivoli Common Reporting license agreement, as shown in Figure 5-27.

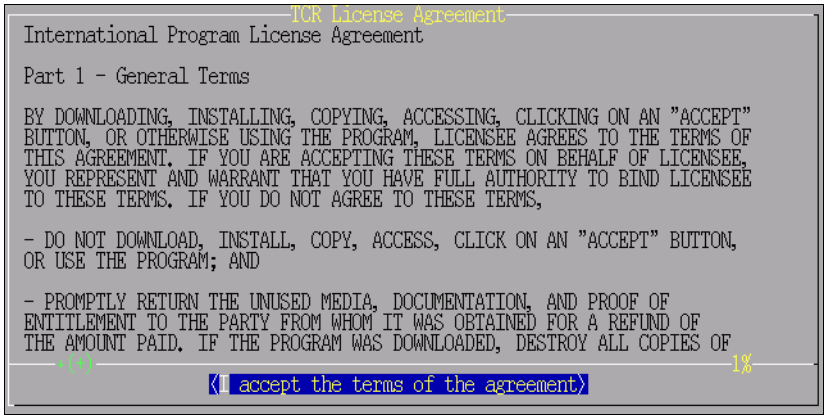

Figure 5-27 Accept Tivoli Common Reporting license agreement

20. The virtual appliance configuration starts, as shown in Figure 5-28. Wait several minutes until the installation completes.

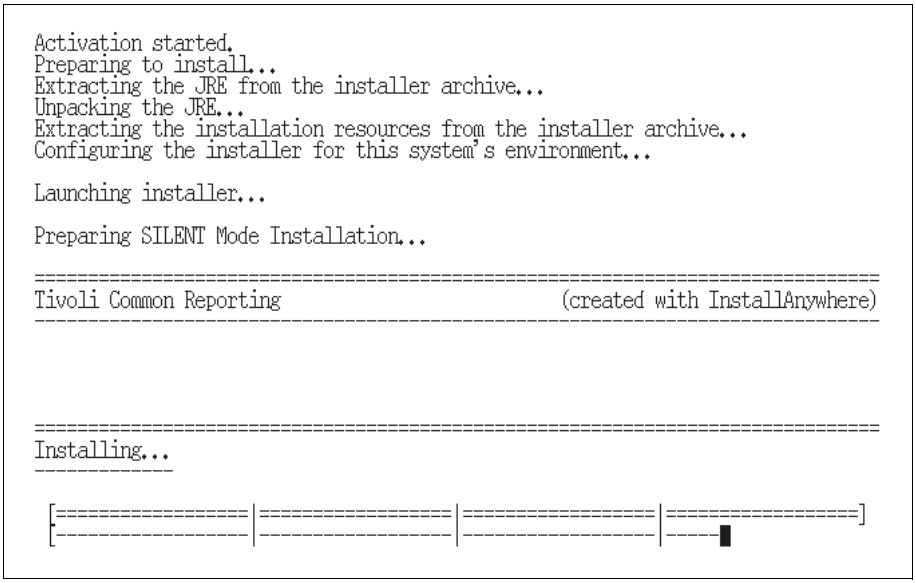

Figure 5-28 Tivoli Common Reporting installation

21. Log on to the server as `virtuser` after the successful installation (Figure 5-29 on page 127).

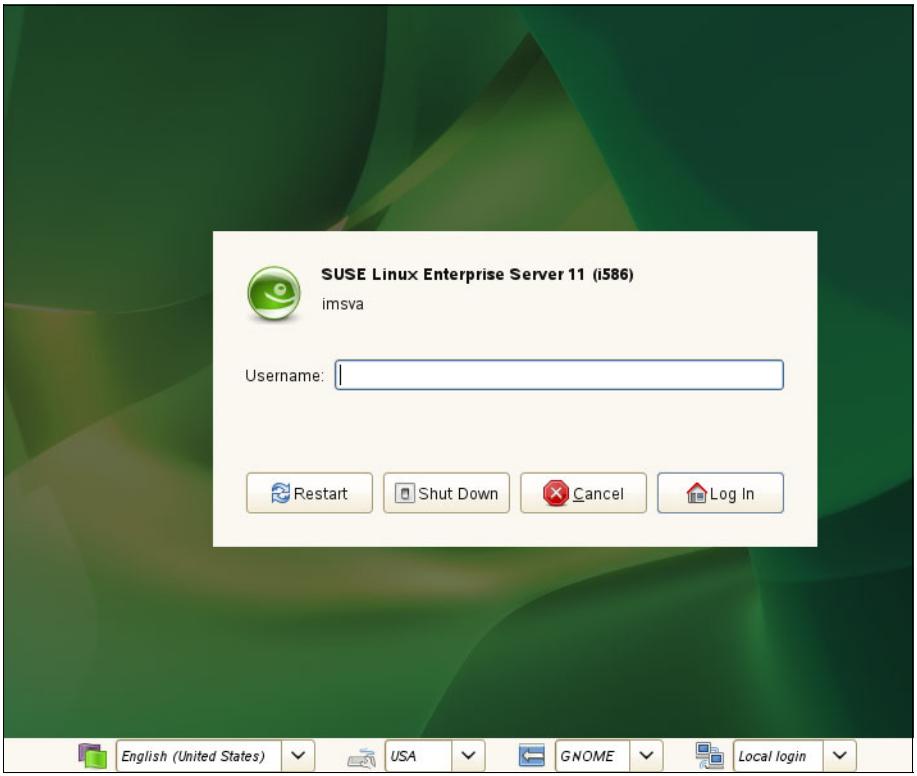

Figure 5-29 IMS Server logon

22. We add the IMS Server manually to the Domain Name System (DNS) on the Windows Domain controller, as shown in Figure 5-30 on page 128.

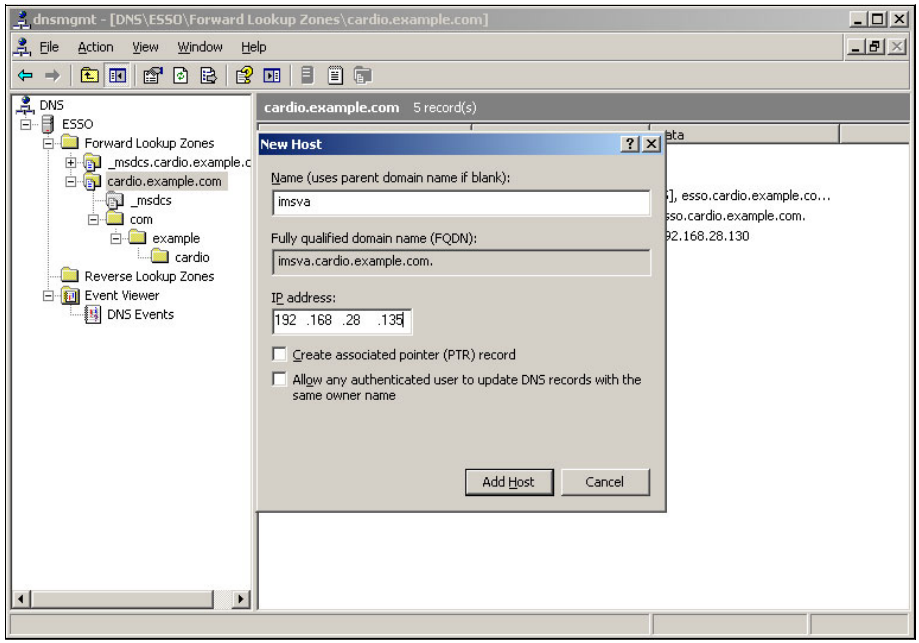

Figure 5-30 Add IMS Server to the DNS

5.2.4 Configuring the database server

The virtual appliance ships without a bundled database. Configuration scripts are included to provision a database on either DB2, Oracle, or MS SQL server. The cardio healthcare company has a DB2 environment that runs on Windows servers and uses the preconfigured SQL scripts to configure DB2 silently. The organization used the database user db2admin to configure DB2. Follow these steps:

1. Set the DB2 command-line environment.

   ```
   db2cmd
   ```

2. Create the database for the virtual appliance on the DB2 server. It is important to use an 8 KB pagesize and UTF-8.

   ```
   DB2 CREATE DATABASE imsvadb AUTOMATIC STORAGE YES  ON 'C:\' DBPATH
   ON 'C:\' ALIAS imsvadb USING CODESET UTF-8 TERRITORY US COLLATE
   USING SYSTEM PAGESIZE 8192
   ```

3. Connect to the newly created database.

   ```
   db2 CONNECT TO imsvadb user db2admin
   ```

The scripts to configure the database are on the virtual appliance in the following directory:

```
/opt/IBM/ISAM_E-SSO/IMS_Server/com.ibm.tamesso.ims-delhi.build.boot/src
/database/data/sql/db2/create-schema
```

Change to this directory within the db2 runtime environment, and run the scripts in the related subdirectories to configure the database:

1. Run the script `init.sql`:

   ```
   db2 -tf "create-schema\init.sql"
   ```

2. Run the script `log.sql`:

   ```
   db2 -tf "create-schema\log.sql"
   ```

3. Run the script `initPL.sql`. The delimiter is “!” for DB2 and “/” for Oracle. Run the following command to pass the delimiter:

   ```
   db2 -td! "create-schema\initPL.sql"
   ```

4. Run the script `boot.sql`. This script is in a different subdirectory.

   ```
   db2 -tf "initialize-prod\boot.sql"
   ```

5. Run the script `create-schema\view.sql`:

   ```
   db2 -tf "create-schema\view.sql"
   ```

Database configuration: The database also can be configured later during the initial browser-based IMS Server configuration, as shown in step 4 on page 131. We suggest though to use this script-based version if you prefer to configure the database by using the command line.

5.2.5 Initial IMS Server configuration

In this section, we configure the IMS Server for initial use.

The initial configuration consists of the following tasks:

- Create the IMS Server database schema (done in the last section by command-line, or optionally done as the first step during the initial IMS Server configuration here).
- Provide the database configuration information.
- Specify the fully qualified domain name of the IMS Server.
- Specify the domain of the enterprise directory to which to connect, and enter the lookup-user name and password.

- Decide whether to synchronize the enterprise directory password and IBM Security Access Manager for Enterprise Single Sign-On password (this option is available only for Active Directory).
- Optional: Define the parameters to enable AccessAssistant and Web Workplace password reset if you want to use those features.
- Restart WebSphere Application Server.
- Assign an enterprise directory user to act as the IMS administrator.

Follow these steps to configure the IMS Server:

1. First, start the IBM Security Access Manager for Enterprise Single Sign-On configuration wizard by opening the URL `http://localhost:9043/front` on the IMS Server that is shown in Figure 5-31.

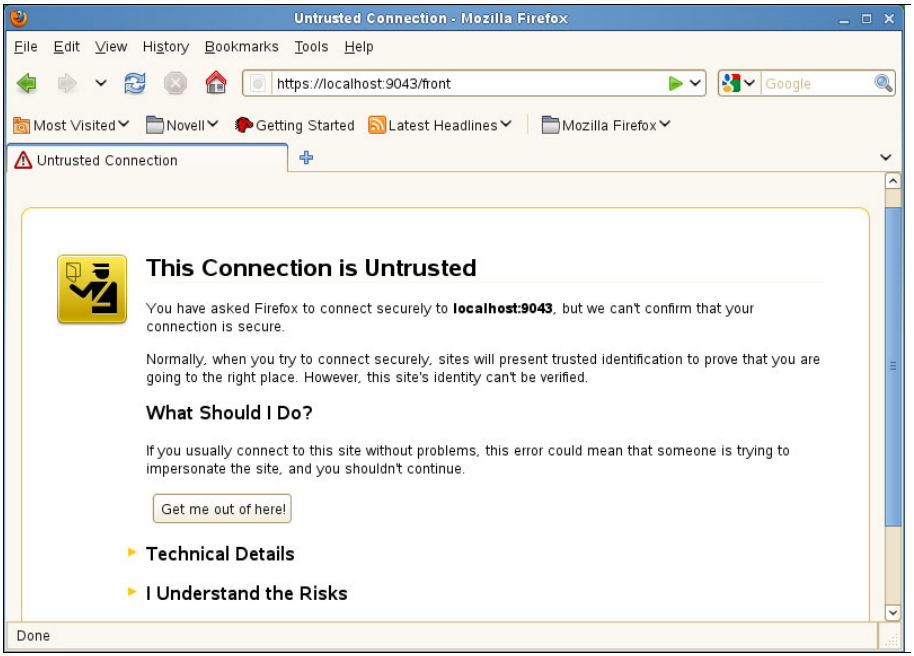

Figure 5-31 Connect to the IMS Server

2. If a certificate warning displays, expand the section **I Understand the Risks** and click **Add Exception**, as shown in Figure 5-32 on page 131.

> **Certificates:** The certificates that ship with the product can be replaced by a company's own trusted certificates.

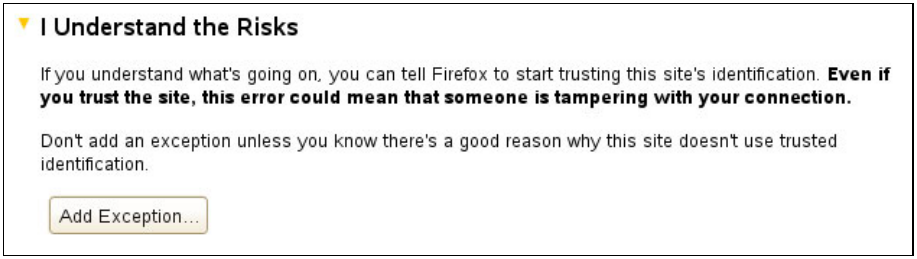

Figure 5-32 Security exception warning

3. Click **Confirm Security Exception** to store the certificate (Figure 5-33).

Figure 5-33 Confirm the security exception

4. The IBM Security Access Manager for Enterprise Single Sign-On Configuration wizard starts. We configured the DB2 database schema earlier and can skip the creation of the IMS Server database schema. Clear the check box, and click **Next**, as shown in Figure 5-34 on page 132.

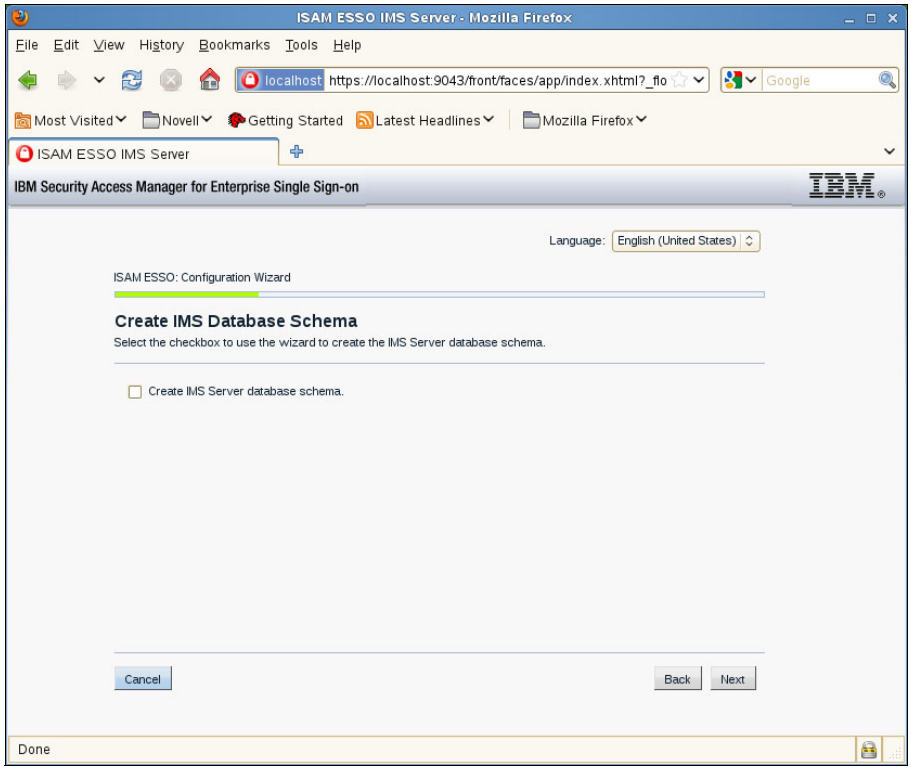

Figure 5-34 Skip IMS database schema creation

5. Choose the database type **DB2 Server**, and click **Next** (Figure 5-35 on page 133).

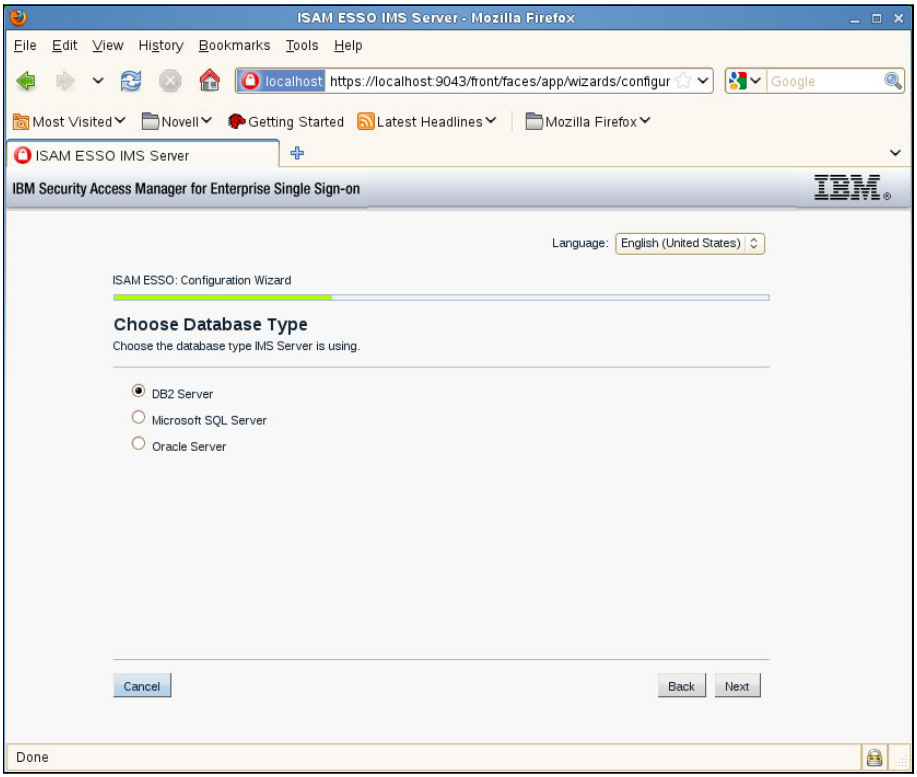

Figure 5-35 Choose database type DB2 Server

6. Enter the database parameters, as shown in Figure 5-36 on page 134, and click **Next**:

Host Name	Fully qualified domain name. The cardio healthcare organization defined `esso.cardio.example.com` as the DB2 server.
Port	Database port. The default port is `50000`.
Database Name	This name is the database that we specified in the previous paragraph. The cardio healthcare company used `imsvadb`.
User Name	The database user specified in the previous paragraph. The default is `db2admin`.
User Password	Password of db2 admin user.

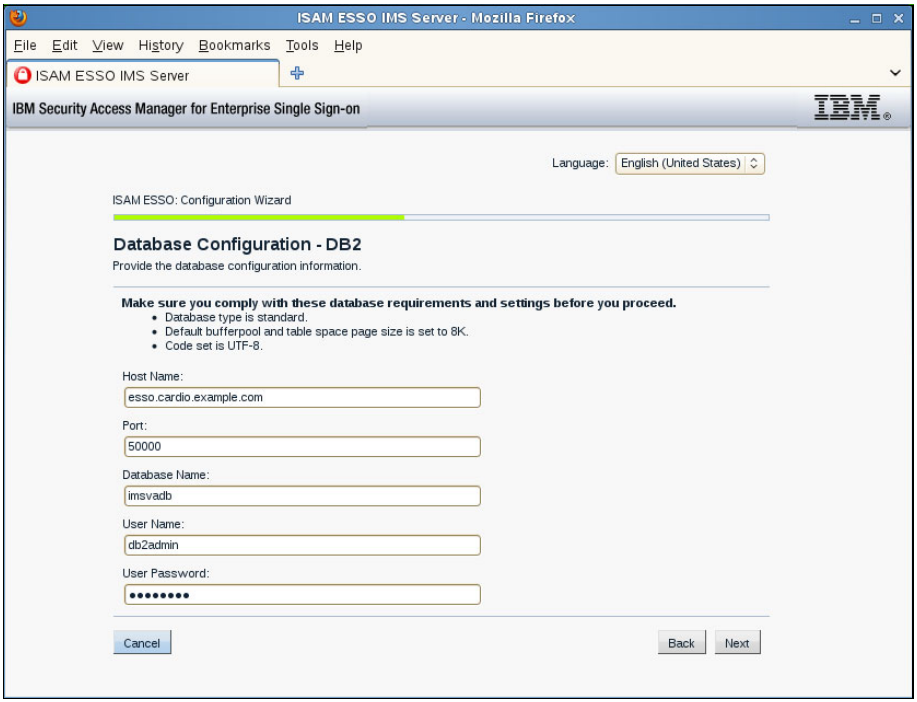

Figure 5-36 Database configuration

7. Enter the fully qualified domain name of the IMS Server, as shown in Figure 5-37 on page 135. The cardio healthcare company created `imsva.cardio.example.com`.

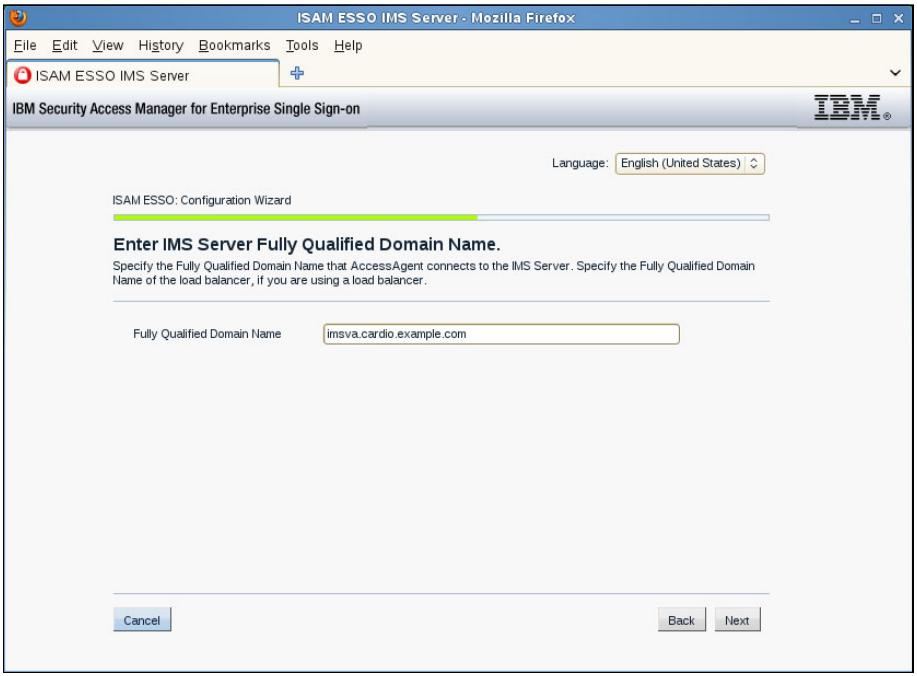

Figure 5-37 Enter the IMS Server fully qualified domain name

8. For the enterprise directory configuration repository listing, the cardio healthcare company decided to add its Enterprise Active Directory as the repository. Click **Add new repository** (Figure 5-38 on page 136).

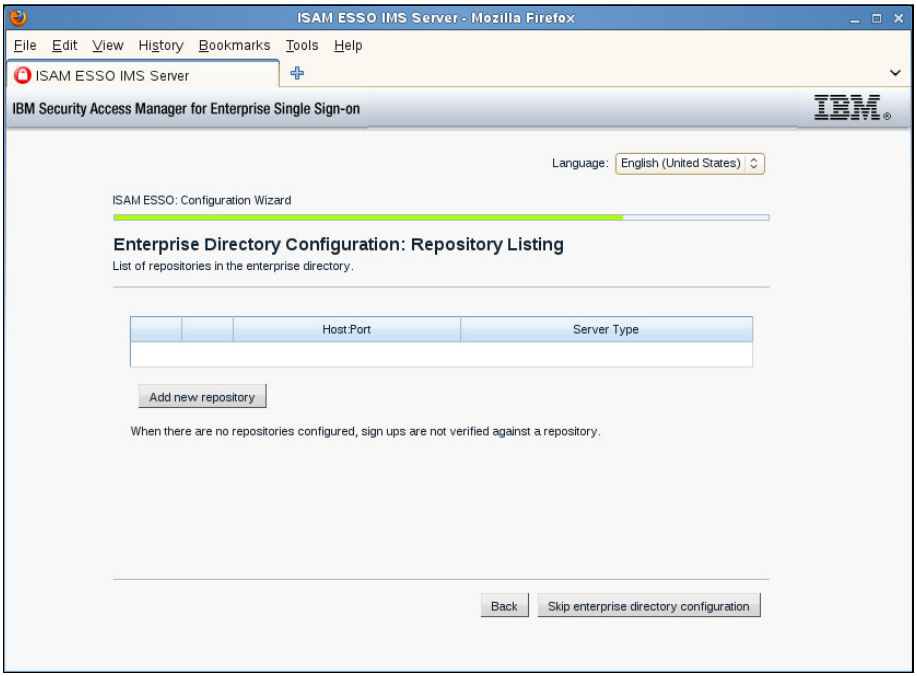

Figure 5-38 Add an Active Directory repository

9. Keep the default setting **Active Directory** as the enterprise directory type, as shown in Figure 5-39 on page 137.

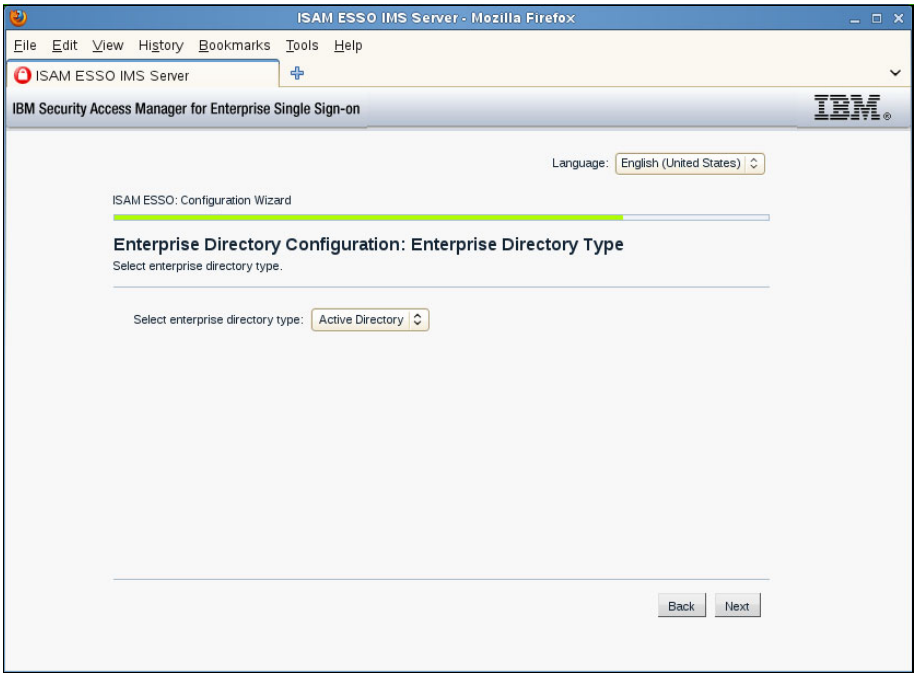

Figure 5-39 Select enterprise directory type Active Directory

10. The organization decided to use password synchronization and changed the default value from No to **Yes** to enable password synchronization (Figure 5-40 on page 138). This step synchronizes the IBM Security Access Manager for Enterprise Single Sign-On user accounts with their Active Directory accounts.

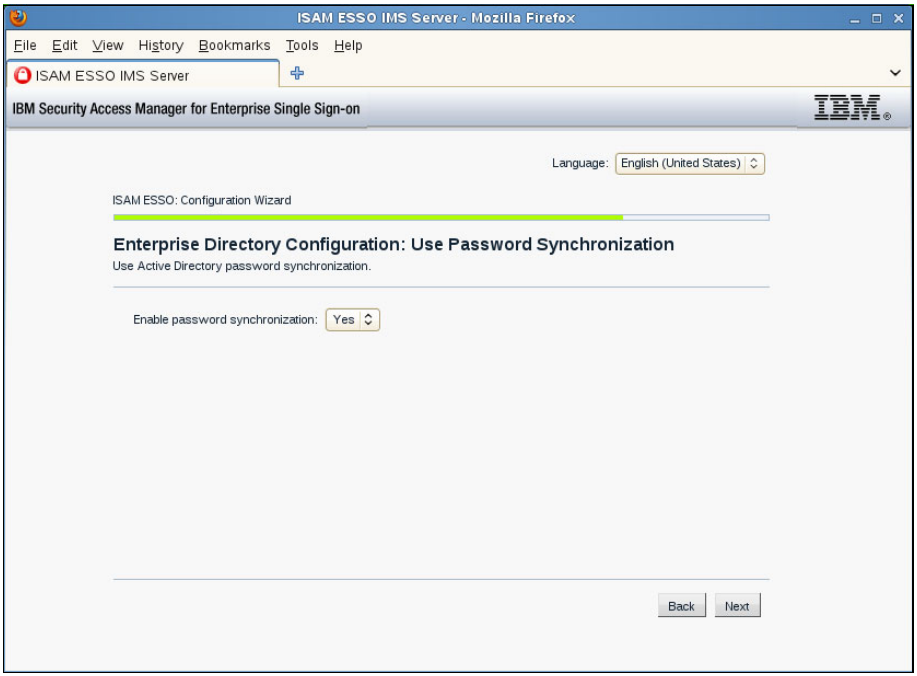

Figure 5-40 Enable password synchronization with Active Directory

11. Specify the parameters for the new repository, as shown in Figure 5-41 on page 139:

Domain Controller FQDN
Fully qualified domain name (FQDN) of the Active Directory (AD) domain controller. The cardio healthcare company used `esso.cardio.example.com` for this name.

Domain DNS name Domain DNS name. The cardio healthcare company used `cardio.example.com` for this name.

Domain NetBIOS name
Domain NetBIOS name. The cardio healthcare environment used `CARDIO`.

Port The cardio healthcare company used port `389`.

Bind user name and password
Define the user name and password of an Active Directory account that has privileges to look up users in the Active Directory used as IBM Security Access Manager for Enterprise Single Sign-On users. The

cardio healthcare company defined the user account `lookup`.

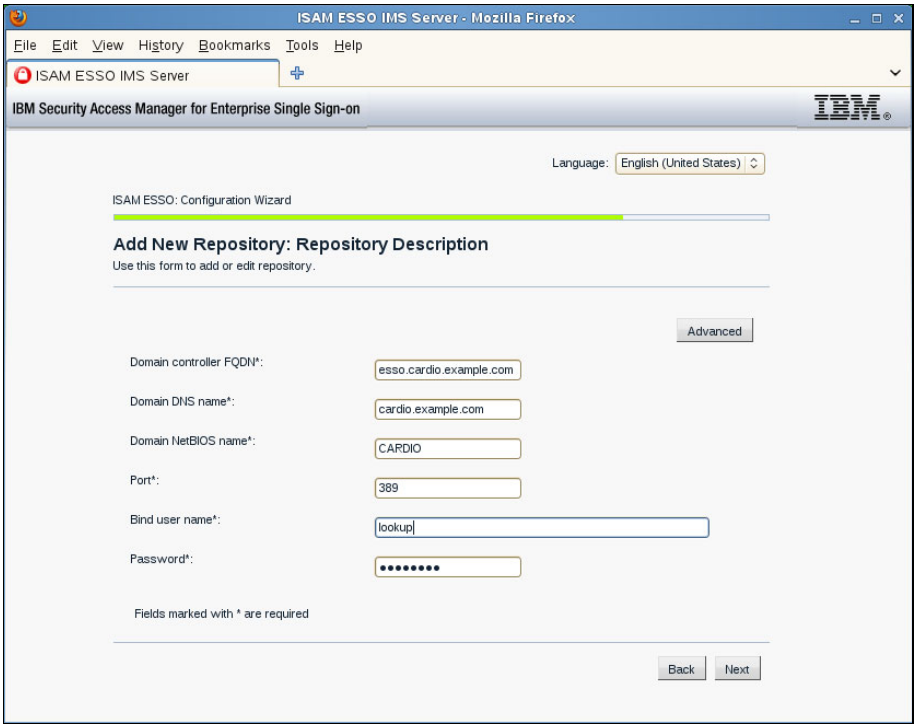

Figure 5-41 Specify repository parameters

12. The cardio healthcare company does not use Secure Sockets Layer (SSL) for Active Directory communication in its test environment. As a result, we need to install the IBM Tivoli Identity Manager Adapter for Active Directory for web services to be able to reset passwords. Click **Yes** for AccessAssistant/Web Workplace password reset. Specify the following parameters, as shown in Figure 5-42 on page 140:

Host The host where the Active Directory adapter for web services is installed. This host must be a Windows system. The cardio healthcare company installed Active Directory on the `esso.cardio.example.com` domain controller.

Use SSL The cardio healthcare company does not use SSL in their test environment. Select **No**.

Port We use the default port `45990`. You can, however, configure this port by using the Tivoli Identity Manager Adapter command-line user interface.

Bind user name The default user name is `agent`.

Simple authentication is enabled
Select `No`.

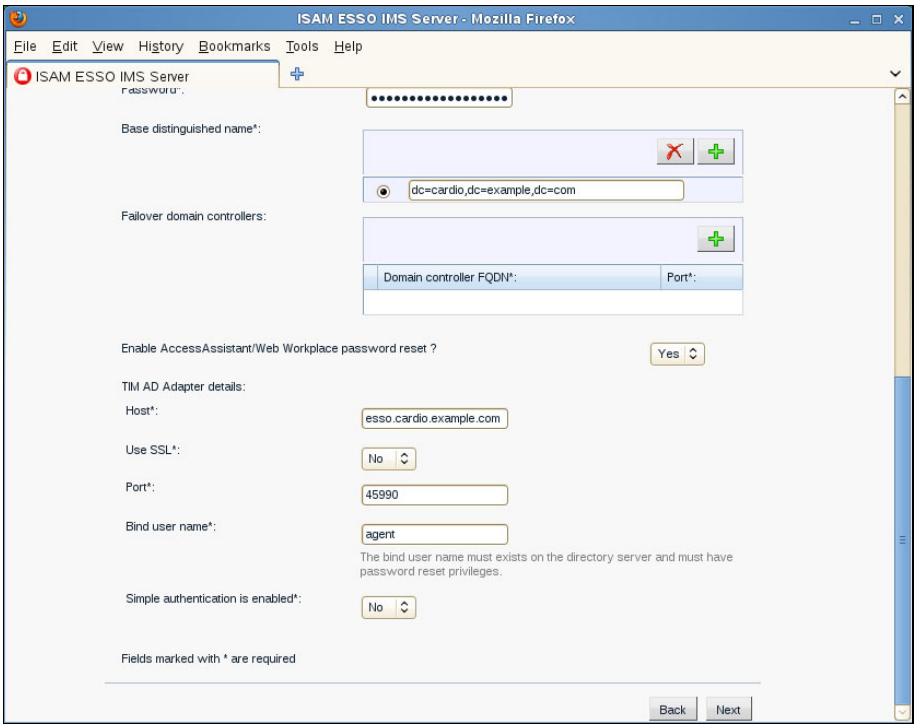

Figure 5-42 Specify connection details for the Tivoli Identity Manager Adapter

Important: The correct choice is **Active Directory adapter for web services** and not Active Directory adapter. These selections are different adapter options.

13. Click **Next** to continue (Figure 5-43 on page 141).

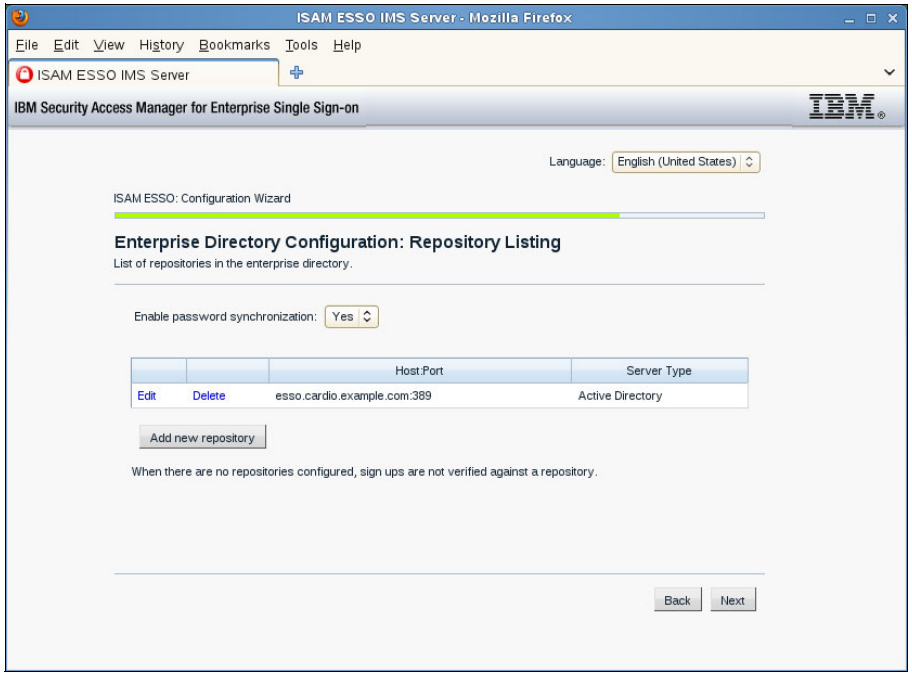

Figure 5-43 Click Next on a successful configuration

14. Confirm the settings and click **Save** to start the IMS configuration, as shown in Figure 5-44 on page 142. This process takes around a minute.

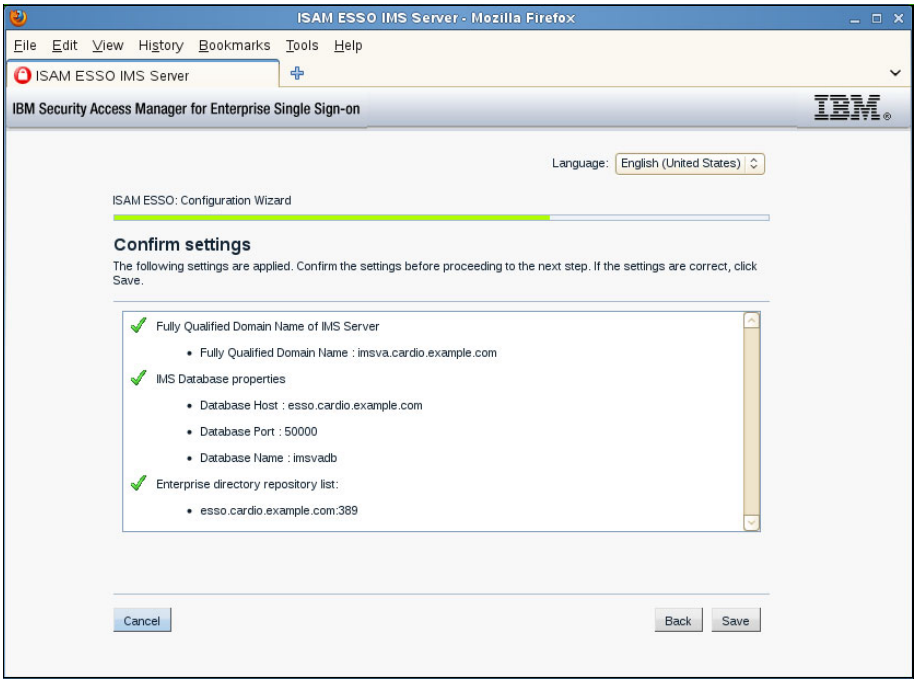

Figure 5-44 Finish IMS configuration wizard

15. After a successful installation, the following window opens, as shown in Figure 5-45 on page 143.

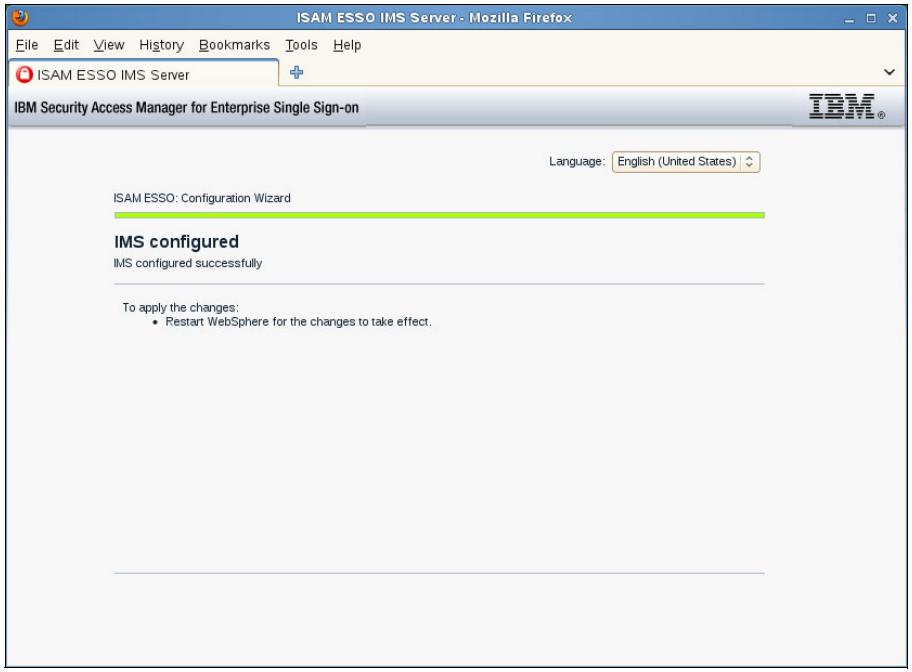

Figure 5-45 Successful IMS configuration

16. Stop and start the WebSphere Application Server after this configuration to apply the changes. The icons are already on the desktop of the virtual appliance (Figure 5-46).

Figure 5-46 Start and stop the WebSphere Application Server

17. Verify the successful WebSphere start or check for errors in the logfile, as shown in Figure 5-47 on page 144.

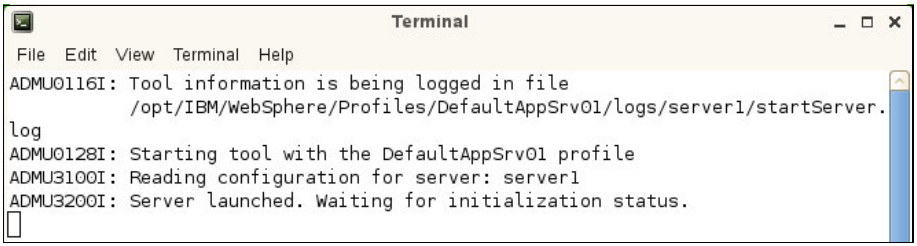

Figure 5-47 Verify WebSphere start

5.2.6 Provisioning an IMS administrator and verifying the installation

Follow these steps:

1. After restarting WebSphere, log on to the IMS Server as `virtuser` at `http://imsva:9043/webconf`, as shown in Figure 5-48.

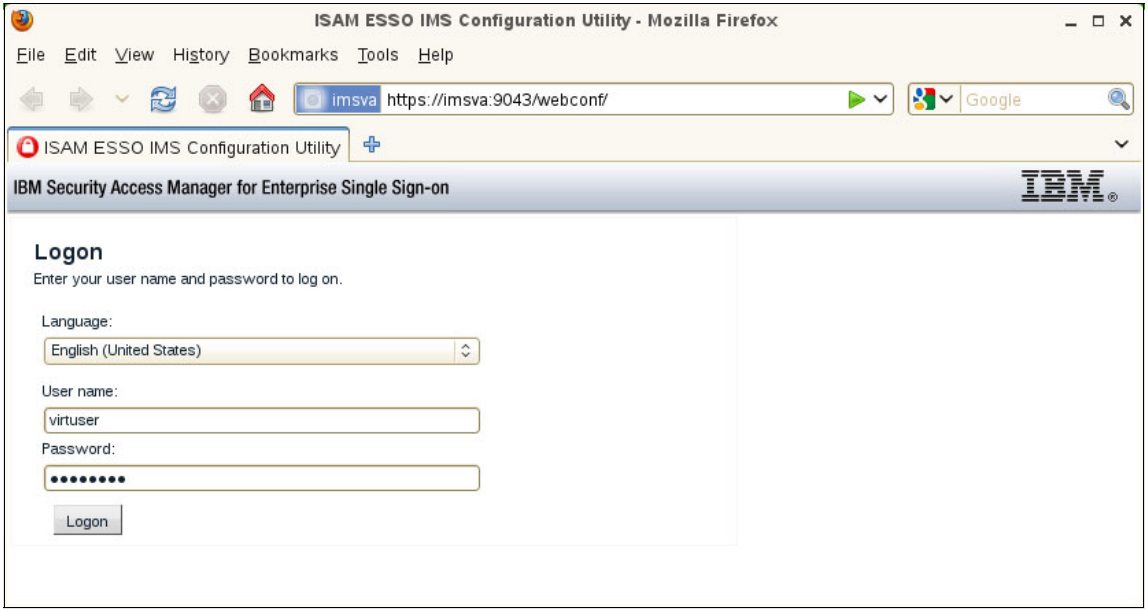

Figure 5-48 Log on to the IMS Server as virtuser

2. On the Welcome window, click **Provision IMS administrator** (Figure 5-49 on page 145).

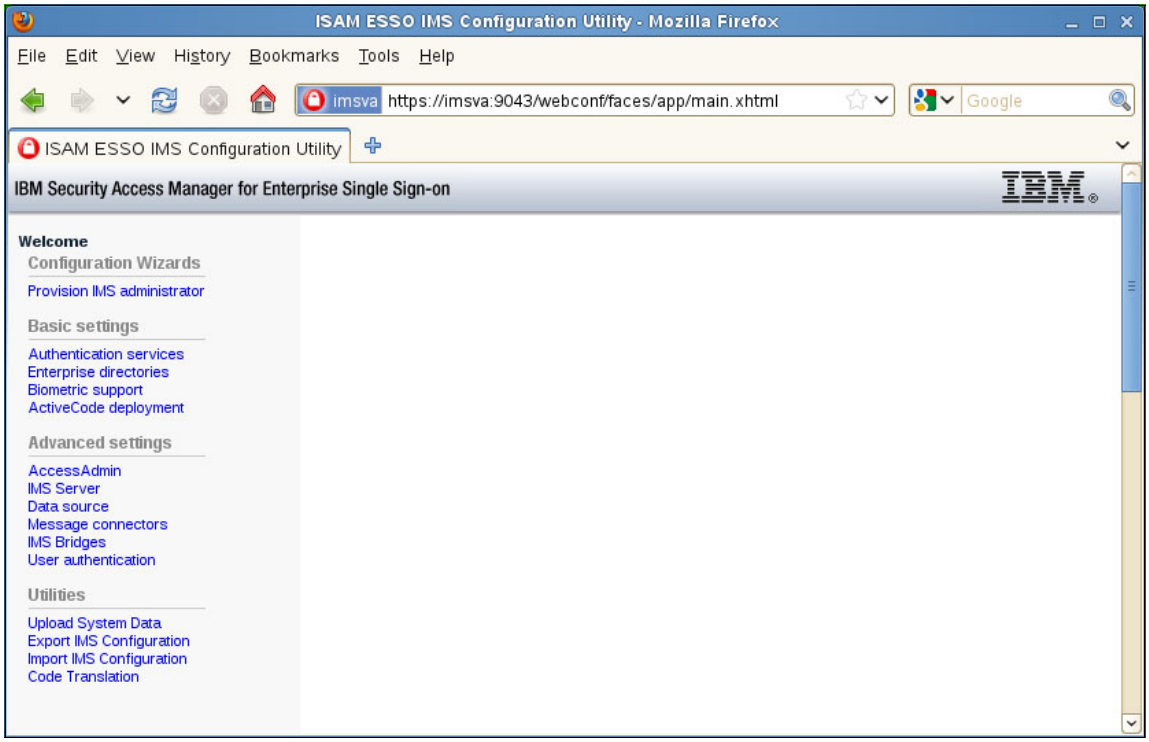

Figure 5-49 Click Provision IMS administrator

3. The cardio healthcare organization created the Windows user imsadmin before. The company provisions this user to be the IMS administrator now by typing `imsadmin`, as shown in Figure 5-50 on page 146.

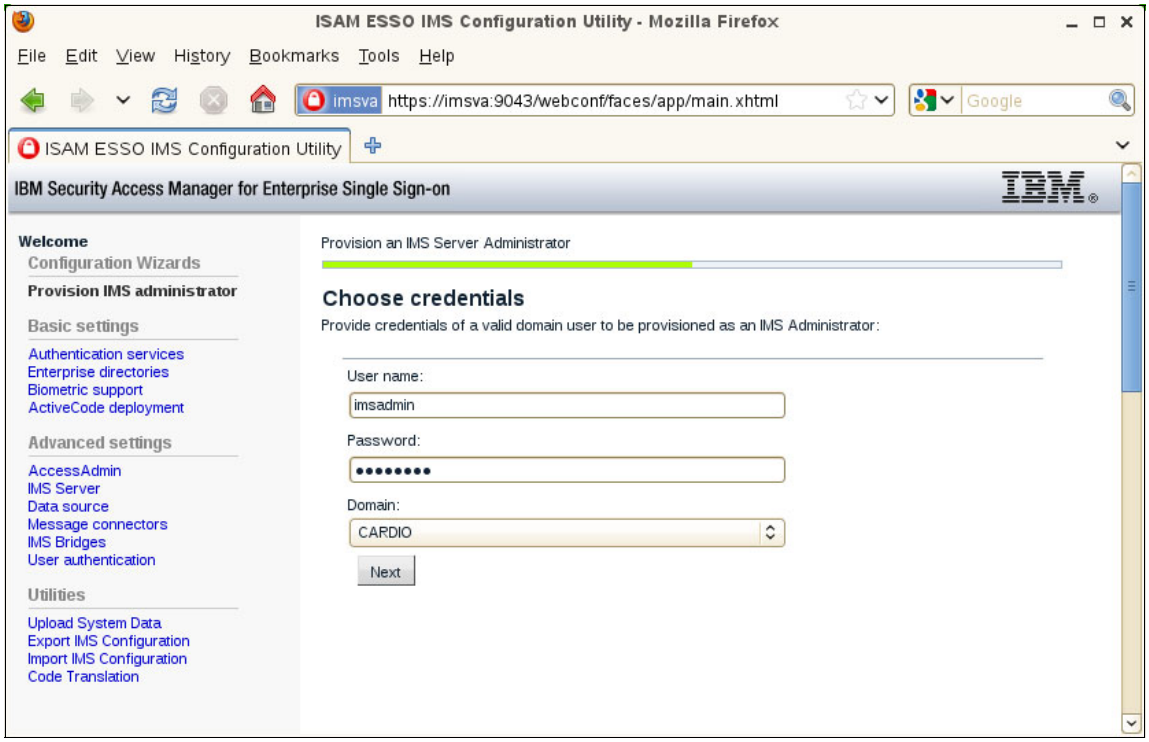

Figure 5-50 Provision new IMS administrator user imsadmin

4. Click **Finish** on the next page (Figure 5-51 on page 147).

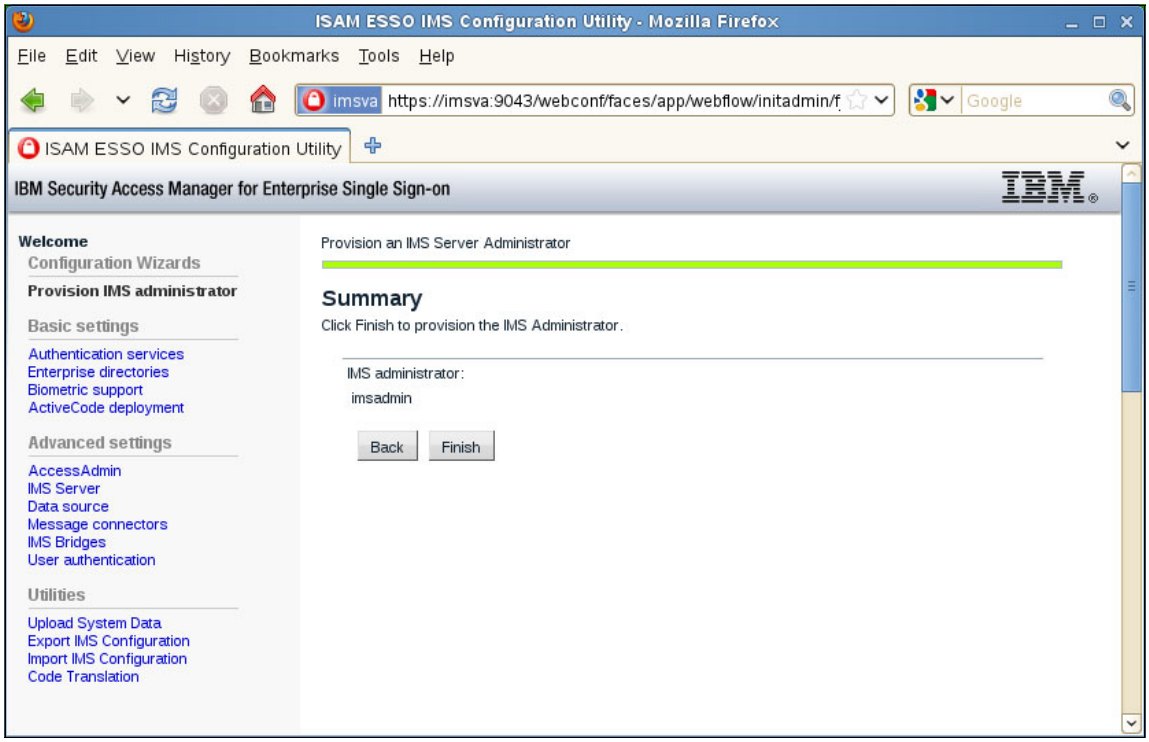

Figure 5-51 Finish IMS administrator provisioning

5. The user imsadmin is provisioned as IMS administrator (Figure 5-52 on page 148).

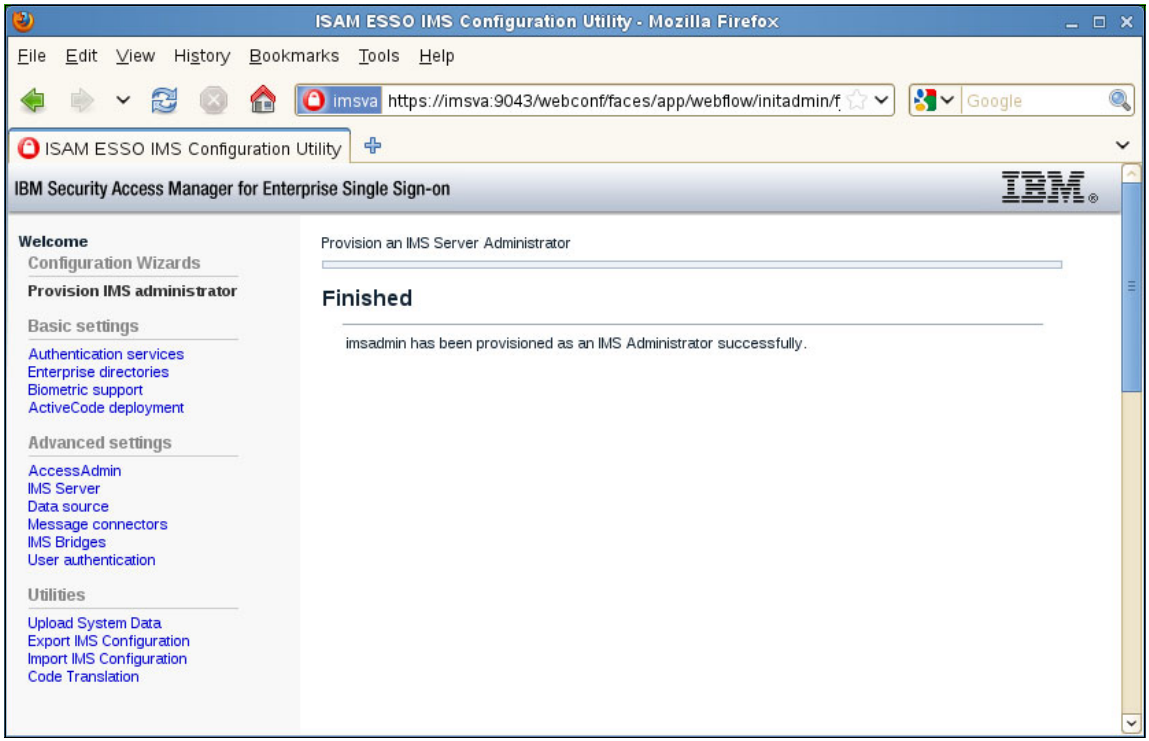

Figure 5-52 Successful IMS administrator provisioning

5.2.7 Configuring user and machine policy templates

Now that we finished the IMS Server base configuration, it is time to configure the user policy template and the machine policy template for personal desktops.

Ideally, the machine policy templates and user policy templates are configured at this time. However, we describe these tasks in Chapter 7, “Strong authentication using RFID” on page 227. We show how you can configure the policies for a personal and a shared desktop at the same time, together with the radio frequency identification (RFID) settings, in one workflow.

Technically, you can install AccessAgents and roll out users first, without the user and machine policy templates in place. If you configure the default machine policy template after you roll out AccessAgent on some workstations, you must apply the machine policies first to the existing machines and then restart the machines.

Machine policy templates: Policies are assigned automatically to new machines and users only, but not to existing machines and users. We suggest that you create the policy templates before adding users and machines to the IMS Server (through users that sign up or the deployment of AccessAgent software).

Manage policies on a policy template level, and not on a machine or user level directly. Create a machine policy and apply the policies to existing machines. If you need to change policies later, change the policies through the machine policy templates again. Do not change policies on machines directly to keep the policies in control.Keep only as few machine policy templates as needed. Do not change values on a single user or machine. Control the policies instead directly on the template level. Change the template, or create a template if needed, and assign it to the correct machines or users.

If you change policies on the machine or user level directly, it can be difficult to track the changes. Also, those values are overwritten when assigning a template to machines.

You can assign the policies in the AccessAdmin. Search for machines by clicking **Machines** → **Search**. Type a search string. Select the “found” machines to which you want to assign a machine policy template. Select from a list of existing machine policy templates. Select **Assign**.

5.2.8 Deploying AccessAgent

The next step in our basic deployment is to install AccessAgent on all workstations that require single sign-on. First, we configure the software package parameters. Then, we show a manual GUI-based installation. Then, we show an automated installation of AccessAgent.

Configuring user, machine, and system policies: In a typical deployment, you now use the AccessAdmin interface to configure the user, machine, and system policies before you install AccessAgent. However, in this book, we describe these steps in 5.4.1, “Managing policies” on page 196.

AccessAgent performs the following primary functions:

- It monitors for applications that are configured for single sign-on and acts on them.
- It communicates with the IMS Server to obtain configuration data and retrieve user Wallets.

- It allows users to access their Wallets and manage their credentials.

The cardio healthcare company reviews several AccessAgent setup parameters. They can be edited by modifying the `SetupHlp.ini` file in the AccessAgent Config installation directory, before running the AccessAgent installer. The `SetupHlp.ini` file contains three categories of parameters:

- Options available only at setup time
- Options that are available at setup and AccessAgent run time that map to multiple registry values each
- Options that are available at setup and AccessAgent run time that map to one registry value each

The options that map to registry values can be modified after the AccessAgent setup, but the options that are only available at setup time cannot be set or changed after the AccessAgent installation. If those options are required post-installation, you must first uninstall AccessAgent, and then reinstall with the setup time only parameters set as needed. Carefully review each option and determine whether it is necessary to modify the values based on your deployment.

For the IBM Security Access Manager for Enterprise Single Sign-On Java Observer module to trigger for Java applications, you must, for instance, specify the paths to the Java virtual machine (JVM) directories installed on the workstation. Instructions for modifying this option are listed in “Modifying the SetupHlp.ini file” on page 150[1].

In the following sections, we examine both the manual GUI-based and the automated silent installation. For both situations, we need to configure the `SetupHlp.ini` file. We first look at modifying the options in the `SetupHlp.ini` file to assist in streamlining the deployment of AccessAgent to multiple workstations that use Windows supported software distribution tools.

Then, we demonstrate both installations, because the cardio healthcare company uses both methods concurrently.

Modifying the SetupHlp.ini file

The cardio healthcare company must modify the `SetupHlp.ini` file with the following options:

1. Copy the entire AccessAgent installation directory to your workstation.

[1] You can also set the paths to the JVMs after your initial installation by using `C:\Program Files\Encentuate\JavaSupport\JSupportInstaller.bat`. See “Chapter 5. AccessAgent setup”, section “Setting automatic sign-on for Java applications” in the *IBM Security Access Manager for Enterprise Single Sign-On Planning and Deployment Guide Version 8.2*, SC23-9952-03.

2. Locate the `SetupHlp.ini` file in the Config directory inside the AccessAgent installation directory. Double-click `SetupHlp.ini` to edit it with a text editor.
3. Modify the address to the IMS Server. Locate the line:

   `ImsServerName=<ISAM ESSO IMS Server>`

   Change the value of `<ISAM ESSO IMS Server>` to the fully qualified host name of the IMS Server:

   `ImsServerName=imsva.cardio.example.com`
4. Because the cardio healthcare company plans to support Java applications in the future, we add the JVM directories. Locate the line:

   `;JVMInstallationDirectories=<JVM Directory 1>|<JVM Directory 2>|<JVM Directory 3>`

   Change the value to include paths to all installed JVMs. It is important to make sure that you remove the semi-colon (`;`) symbol from the start of the line and to separate each of the directories with the vertical bar (`|`) symbol:

   ```
   JVMInstallationDirectories=C:\Program
   Files\Java\jre1.6.0_01|C:\Program Files\Java\jre1.5.0_03
   ```
5. Save and close `SetupHlp.ini`.

Optional configuration parameters in the SetupHlp.ini file

To better understand optional configuration options for the `SetupHlp.ini` file, the cardio healthcare company looks closely at some additional and important parameters:

- `FirstSyncMaxRetries`

  System data, such as machine policies and AccessProfiles, are downloaded and stored as a machine Wallet during the initial start after the AccessAgent installation (default location: `machine.wlt` and `machine.wlt.bak` in `C:\Program Files\IBM\ISAM ESSO\AA\Cryptoboxes\Wallets`).

  If the download was unsuccessful with Version 8.1 at start-up, the synchronizer retried every 20 seconds for three times. Afterward, the synchronizer retries at the periodic synchronization interval as defined globally on the IMS Server (typically, 30 minutes).

  Use the new configuration parameter `FirstSyncMaxRetries` to change the number of attempts to retry if the first synchronization fails during installation. The default value is `3`. Set this value to a higher number if the IMS Server can be reached only after a virtual private network (VPN) connection to the company's intranet was established. This connection can take more than 1 minute.

Machine Policies setup: For some machine policies to be effective, they must be set before start-up. Or, the default installer is used, and two start-ups might be required instead.

- `FirstSyncRetryIntervalMins`

  `FirstSyncRetryIntervalMins` defines the time interval between each retry attempt by minutes during the installation.

- `RemoveWallet`

  This parameter defines whether to remove the Wallet during AccessAgent uninstallation (1 YES|0 NO). Setting this value to 1 removes the Wallet during uninstallation. This parameter can be useful for reinstallations when Wallet files can block a new installation.

- `DisableWin7CAD`

  This parameter determines whether to disable Ctrl+Alt+Del in Windows Vista and Windows 7.

- `AAInstallDir`

  This parameter specifies the AccessAgent installation base directory. The default value for V8.2 is:

  ```
  AAInstallDir=C:\Program Files\IBM\ISAM ESSO\AA\
  ```

Manual installation of AccessAgent

First, the cardio healthcare company examines the manual installation method for AccessAgent:

1. From the AccessAgent installation directory, double-click the `Setup.exe` to begin the AccessAgent installation.

Version: At the time of writing this book, the AccessAgent version is 8.2.0.0501.

2. At the InstallShield Wizard window, select the installation language and click **OK**, as shown in Figure 5-53.

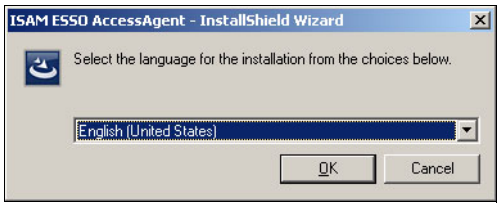

Figure 5-53 AccessAgent - Installation wizard

3. Currently, 19 languages are supported. Each language has a different `.msi` file in the installation directory. We show in the next paragraph how to install AccessAgent silently, by using a specific language setting. We select **English [United States]**, as shown in Figure 5-54, and click **Next**.

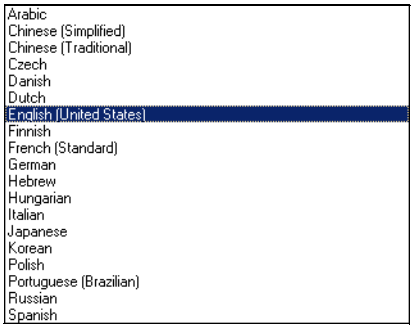

Figure 5-54 Support languages for AccessAgent

4. At the Welcome window, click **Next** (Figure 5-55 on page 154).

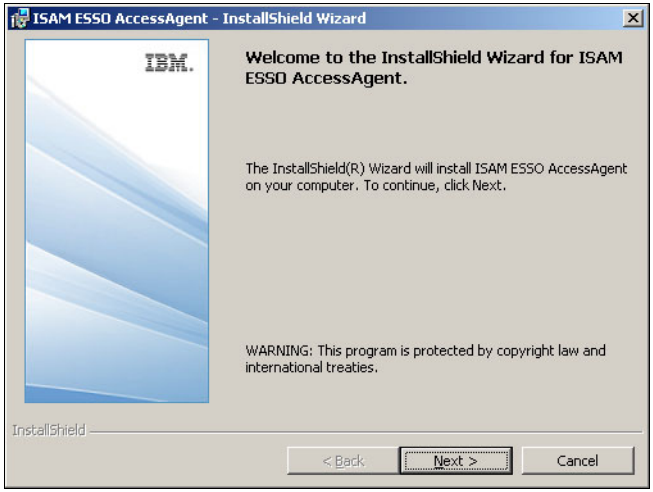

Figure 5-55 Welcome window

5. Select the product for which you have a license and click **Next**. The cardio healthcare organization purchased the IBM Security Access Manager for Enterprise Single Sign-On Suite, as shown in Figure 5-56.

Figure 5-56 Product License Validation

6. Read the license agreement, and if you accept the terms, select **I accept the terms in the license agreement**, and click **Next,** as shown in Figure 5-57 on page 155.

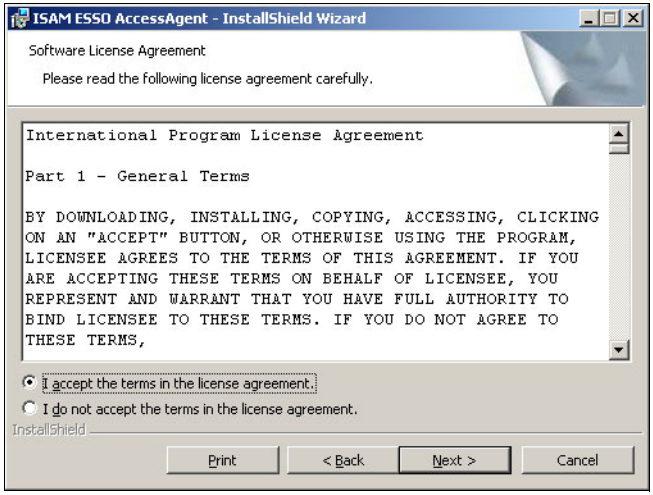

Figure 5-57 License Agreement

7. Keep the default installation directory, and click **Next** (Figure 5-58).

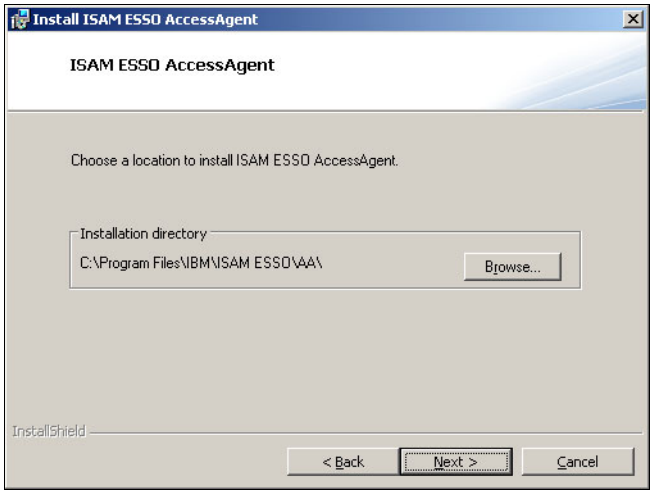

Figure 5-58 Choose the installation directory

8. Click **Install** to begin the installation (Figure 5-59 on page 156).

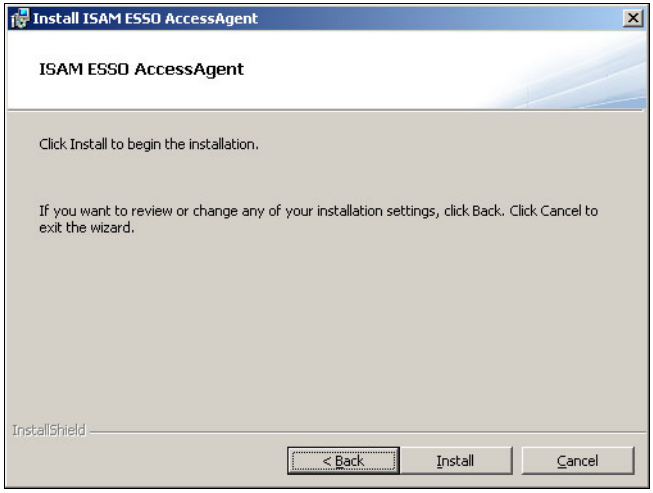

Figure 5-59 AccessAgent Review window

9. AccessAgent installation begins. After the installation completes, the successful installation window opens. Click **Finish**, as shown in Figure 5-60.

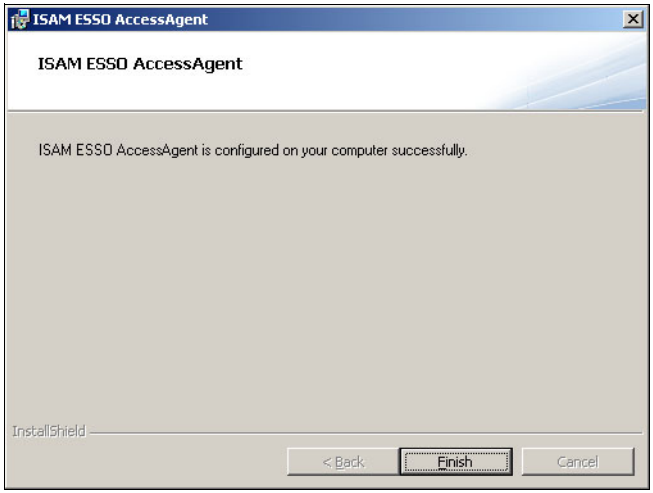

Figure 5-60 AccessAgent installation finished

IMS Server connection: If, during software installation testing, you are prompted to enter the IMS Server connection information, the IMS Server address that was entered in `SetupHlp.ini` is probably incorrect. This message can also appear if the IMS Server was unreachable by the client at installation time.

10. You are now asked to restart the system. Click **Yes** to restart now (Figure 5-61). You can also configure `Setuphlp.ini` to hide this prompt and not force the user to restart immediately.

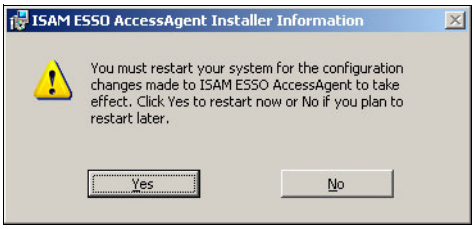

Figure 5-61 Restart after AccessAgent installation

Important: Sometimes, two restarts can be necessary, because after the first restart, AccessAgent synchronizes with the IMS Server and retrieves the machine policies. Certain machine policies require a restart before they can take effect.

AccessAgent installation is complete.

Silent installation of AccessAgent

The cardio healthcare company examines the silent installation method for AccessAgent. Instead of executing the `setup.exe` file, the Microsoft Windows Installer needs to run the `ISAM ESSO AccessAgent.msi` file directly.

Currently, AccessAgent can be installed in 19 languages. A complete list is in the product documentation. The cardio healthcare organization decided to support English as the corporate language. We show how to install AccessAgent silently.

The Microsoft Windows installer is executed by starting `msiexec.exe`. We pass the following switches to the command line:

1. `/i <patch to the MSI file>`
2. `TRANSFORMS=<Microsoft Locale ID>.mst` for the language setting in decimal form, for example, `1033`, for English United States.

3. `/norestart` is optional, if the AccessAgent installer must not automatically restart after installation.
4. `/qn` sets the user interface level to **n- no user interface** for silent installation without user interaction.
5. `/l <logfile>` sets the path to the AccessAgent installation logfile. Optional: The loglevel can be set to a higher value for debugging, for example, `/lv` for verbose logging. See more details by running:

   ```
   msiexec /?
   ```

6. `PATCH="<path to AccessAgent patch>.msp"` is optional, if an AccessAgent patch must be installed directly with the AccessAgent installer.

Software distribution mechanism: All directory paths might need to be set as absolute if a software distribution mechanism was used, for example:

```
/i "c:\swd\ISAMESSO\ISAM ESSO AccessAgent.msi"
```

The cardio healthcare organization installs AccessAgent silently with this command:

```
msiexec.exe /i "ISAM ESSO AccessAgent.msi" TRANSFORMS="1033.mst"
/norestart /qn /l AA82.log
```

5.2.9 Interacting with AccessAgent

After we install AccessAgent, we show the basic functions. In this section, we describe how you sign up for an IBM Security Access Manager for Enterprise Single Sign-On account and modify how you log on to Windows.

The Windows Logon window is different, as shown in Figure 5-62 on page 159.

Figure 5-62 AccessAgent Logon window

Several options changed from the default Windows logon window. You can now select the following options:

- Log On

  Selecting this option logs you on by using your IBM Security Access Manager for Enterprise Single Sign-On password. If you are not yet signed up, you are prompted to sign up. This option, which can be changed, is the default behavior.

- Sign Up

  Signing up creates your IBM Security Access Manager for Enterprise Single Sign-On account and password. It prompts you to provide an answer to a challenge question of your choice.

- Reset Password

  Reset your IBM Security Access Manager for Enterprise Single Sign-On password. How the password gets reset depends on the configuration in AccessAdmin. Options include responding to a challenge question, entering an authorization code that is provided by the help desk, or both.

- Go to Windows to log on

  If you do not want to log in to IBM Security Access Manager for Enterprise Single Sign-On, you can bypass it and log on to Windows. If you choose this logon, AccessAgent does not know who you are, and single sign-on functionality is disabled until you log in to AccessAgent.

The cardio healthcare company plans to educate its users on the steps to expect. In the following steps, we sign up as a user to use IBM Security Access Manager

for Enterprise Single Sign-On and modify the action to take when the credentials are injected into the Windows Logon window:

1. Click **Sign Up**, and enter your domain User name and Password (Figure 5-63). We use the user name `Administrator` and the Windows Domain is `cardio`.

Figure 5-63 Sign up a new user

If you clicked **Log On** in the previous step, but the User name is not registered on the IMS Server yet, you are asked to sign up. Click **Yes**, as shown in Figure 5-64.

Figure 5-64 Sign up new user

2. Next, you are asked to enter your secret answer to a question of your choice. Select a question and enter an answer that you are not going to forget. Click **Finish** (Figure 5-65).

> **Questions:** You can customize the questions presented for the secret in **AccessAdmin** → **System Policies** → **Sign up Policies**. You can also specify that a secret is not required at sign-up.

Figure 5-65 Question and secret answer

3. Next, you are prompted for your Windows credentials. We typed User name `Administrator` (Figure 5-66) and selected logon domain **cardio**.

Figure 5-66 Store the Wallet on the workstation after the logoff

4. After the logon, you are prompted whether you want to store the password for the Windows logon into the Wallet, as shown in Figure 5-67 on page 162.

Figure 5-67 Store the password for the Windows logon in the Wallet

5. From now on, the user name and password are injected automatically into the Windows user name and password fields at logon. However, you are still required to click **OK** to continue the logon process. In the next steps, we change the click action so that it is automatic (Auto-logon). For now, click **YES** to save the password in the Wallet.
6. Observe the AccessAgent icon in the system tray, as shown in Figure 5-68.

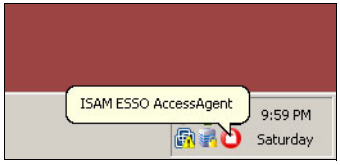

Figure 5-68 AccessAgent tray icon

7. Right-click the AccessAgent tray icon, and then select **Manage Wallet**, as shown in Figure 5-69.

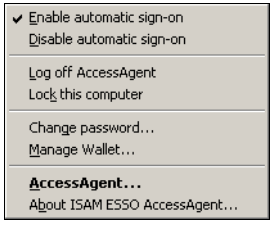

Figure 5-69 AccessAgent tray menu

8. Alternatively, you can double-click the Access Agent tray icon to see the session info:
   - AccessAgent software version.
   - IMS Server location.
   - Last sync time of AccessAgent with the IMS Server. This information is useful for the internal support personnel if the user experiences synchronization issues with AccessAgent.

Now, select **Manage Wallet**, as shown in Figure 5-70.

Figure 5-70 AccessAgent session information

9. Click the drop-down list under Password Entry and select **Automatic logon**. Then, click **Close** (Figure 5-71 on page 164). Also, this behavior can be set globally in the AccessAdmin for the authentication service.

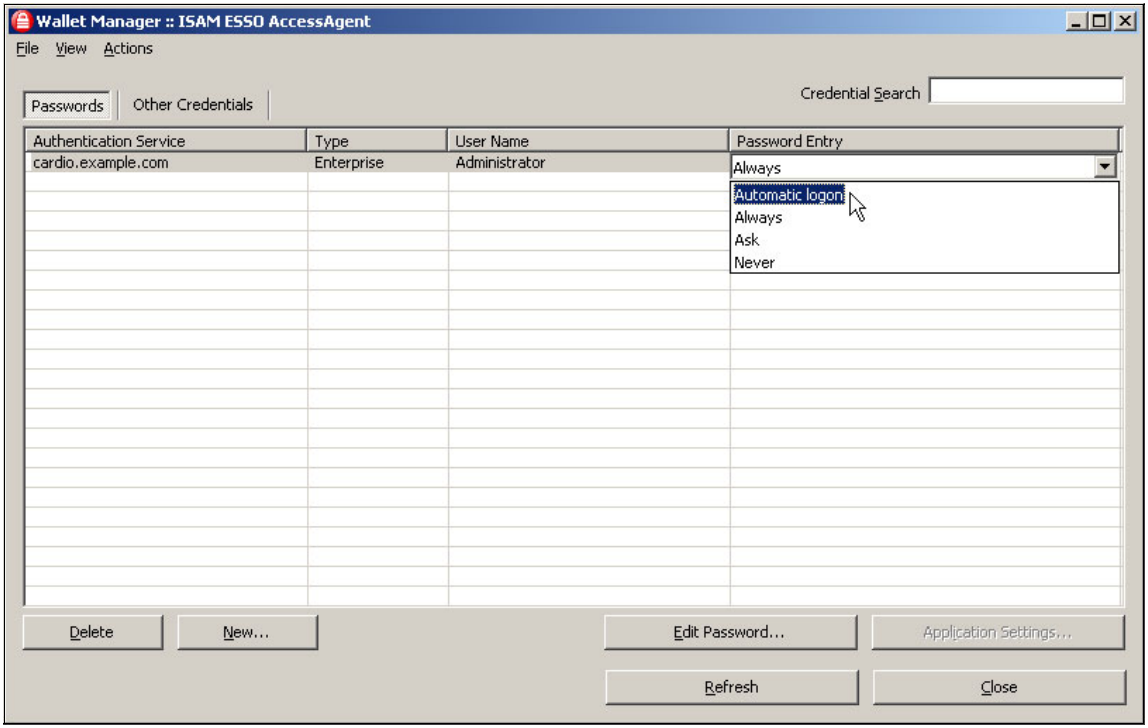

Figure 5-71 Change logon action for the Windows authentication service

10. Log off Windows and log back in with the same user. Notice that AccessAdmin automatically logs you in.

> **Automatic logon:** You can achieve the same application behavior without user interaction by configuring the profile for automatic logon in AccessStudio, the application, and the authentication service.

5.2.10 Installing AccessStudio

AccessStudio is used by administrators only to create AccessProfiles that contain instructions for handling automation for a specific application, such as a Lotus Notes email client or an SAP GUI client. After a profile is created, it can be uploaded to the IMS Server to publish the data for use in the corporate environment.

The installation of AccessStudio is a short process. Follow these steps:

1. Locate the AccessStudio installation binaries and double-click **`setup.exe`**.

2. Select the **Installation language**, as you did for AccessAgent, and click **Next**.
3. Select the product for which you have a license and click **Next**. For the AccessAgent plug-ins, the cardio healthcare organization has a license for the IBM Security Access Manager for Enterprise Single Sign-On Suite. Select **IBM Security Access Manager for Enterprise Single Sign-On Suite**, and click **Next** to install the suite, as shown in Figure 5-72.

Figure 5-72 AccessStudio Product License Validation

4. You are presented with the License Agreement window. If you agree to the terms, select **I accept the terms in the license agreement** and click **Next**. The installation of AccessStudio begins (Figure 5-73 on page 166).

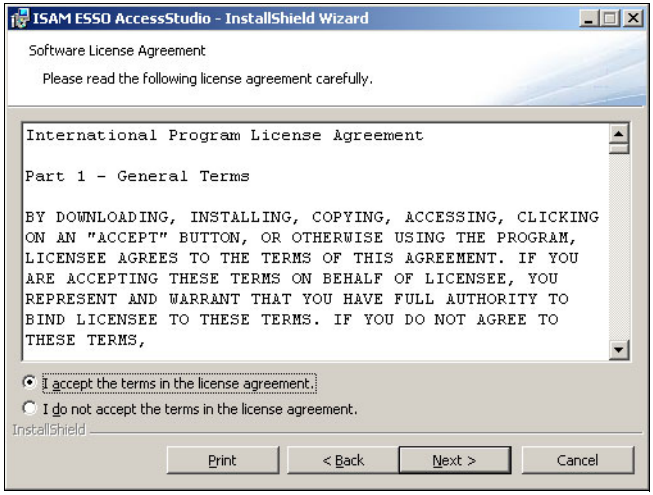

Figure 5-73 AccessStudio License Agreement

5. Select the destination folder and click **Next**, as shown in Figure 5-74. On the next window, verify the information and click **Next** again to start the installation.

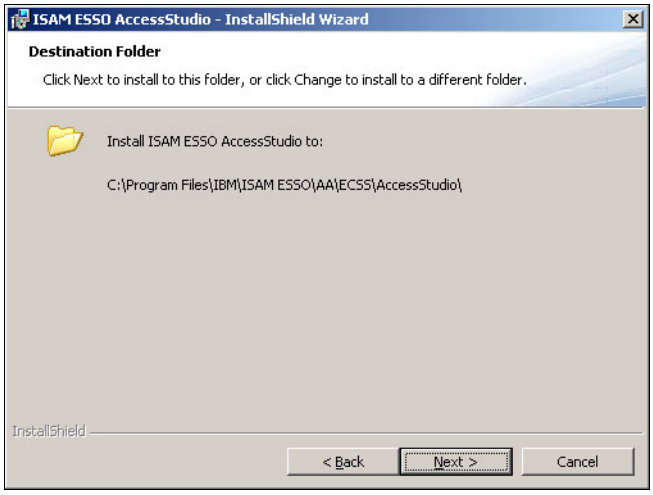

Figure 5-74 AccessAgent Destination Folder

6. After the installation wizard completes, click **Finish**.
7. You can now start the AccessStudio by clicking **Start** → **Programs** → **ISAM ESSO AccessStudio** → **AccessStudio**, as shown in Figure 5-75 on page 167.

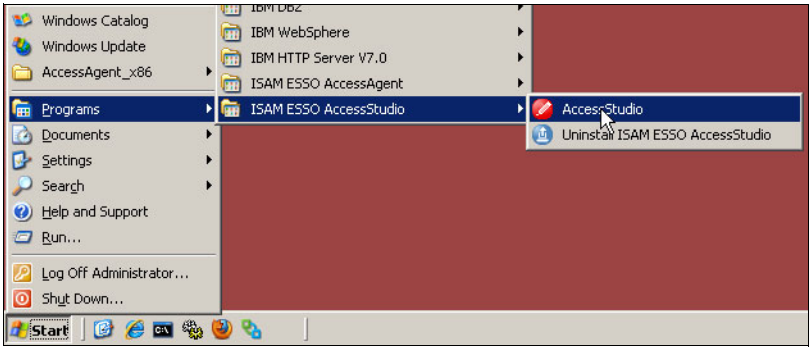

Figure 5-75 Start AccessStudio

8. In AccessStudio, open **File** → **Import data from IMS** (or from the local AccessAgent, in our case). In Figure 5-76, you see the AccessProfiles that ship with IBM Security Access Manager for Enterprise Single Sign-On.

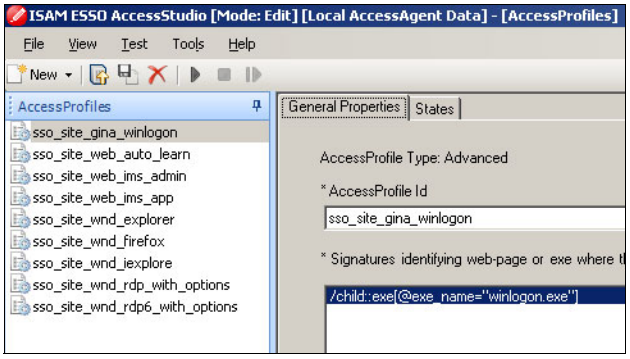

Figure 5-76 AccessProfiles that ship with the product

5.3 Configuring AccessProfile

After we install and configure the base components, we must configure IBM Security Access Manager for Enterprise Single Sign-On so that users can use the single sign-on functionality for their applications. The cardio healthcare company users use three primary applications that we configure for single sign-on:

- Lotus Notes (email)
- SAP client
- VMware View Client

We use the following high-level steps, which are the same steps for any enterprise application, to configure the applications for single sign-on:

1. Configure an authentication service if an authentication service does not exist.
2. Configure an AccessProfile if an AccessProfile does not exist.
3. Move the application profile to the *enterprise authentication services*.
4. If you use more than one user template, define which applications are assigned to each user template.

Understanding the relationships among AccessProfiles, applications, and authentication services is important. AccessProfiles consist of an authentication service and the logical reference to an application. The application is a logical reference to an `.exe` file or a website. An authentication service defines how user credentials are submitted to the application. Multiple applications can use the same authentication service. For more information, see *IBM Security Access Manager for Enterprise Single Sign-On AccessStudio Guide Version 8.2*, SC23-9956-03.

Authentication services can be configured as either an *enterprise authentication service* or a *personal authentication service*. Administrators can change a service to be personal or enterprise through AccessAdmin.

Enterprise authentication services allow greater administrative control over the user interaction with the service. Users are not allowed to delete an enterprise service account from their Wallet, and they cannot set *Never* as an option for the password entry.

Auditing: Audit logs are stored and generated on the IMS Server only for *enterprise authentication services*. They are not generated for *personal authentication services*.

Personal authentication services allow the user more control over how the user wants AccessAgent to interact with the authentication service. Users can have an unlimited number of accounts per service; there is no ability for administrators to grant or deny access to specific users. The administrator can disallow all personal authentication services, but not specific personal authentication services.

Corporate-related authentication services: For all corporate-related authentication services, we advise setting them to *enterprise authentication services* because of the enhanced administrative control and the audit logging.

In the following sections, we configure three applications for single sign-on. Each application involves a different level of effort.

5.3.1 IBM Lotus Notes application

The cardio healthcare company uses Lotus Notes for its corporate email application. In AccessStudio, open **File** → **Import data from IMS** (or from the local AccessAgent, in our case). As shown in Figure 5-77, IBM Security Access Manager for Enterprise Single Sign-On ships with only a few AccessProfiles. IBM Security Access Manager for Enterprise Single Sign-On does not ship with an authentication profile and application profile pre-configured for Lotus Notes.

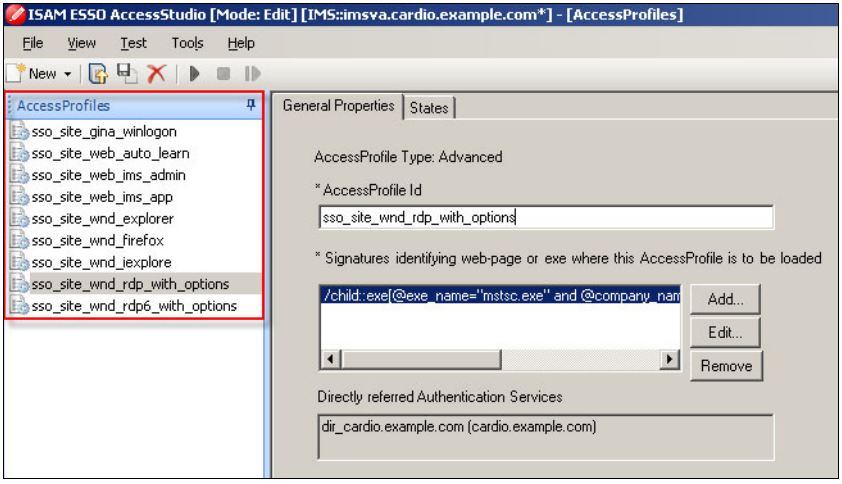

Figure 5-77 Default IBM Security Access Manager for Enterprise Single Sign-On profiles

However, many application profiles are available and supported by IBM. You can download the latest profiles from the IBM support portal website. There is a supported AccessProfile for Lotus Notes available from that website, so there is no need to use AccessStudio to create an AccessProfile for this application.

The latest profile for Lotus Notes can be downloaded from the following website:

`https://www.ibm.com/support/docview.wss?uid=swg24029132`

This Enterprise Software Profile Bundle release contains AccessProfiles and the relevant documentation for the following enterprise applications:

- IBM Cognos® 8 Business Intelligence
- IBM FileNet
- IBM Lotus Notes
- IBM Lotus Sametime® Connect

- IBM Personal Communications
- SAP GUI for Windows
- Oracle E-Business Suite

We use the Lotus Notes profile as an enterprise authentication service. We configure additional policy settings for the Lotus Notes authentication service as described in the following steps.

Follow these steps to configure email settings:

1. Log in to AccessAdmin as an administrative user by using the URL `https://imsva/admin/faces/auth/login.xhtml` where *imsva* is the IMS Server name.
2. On the left panel under System, click **Authentication service policies** (Figure 5-78). Lotus Notes is in the Enterprise authentication services section.

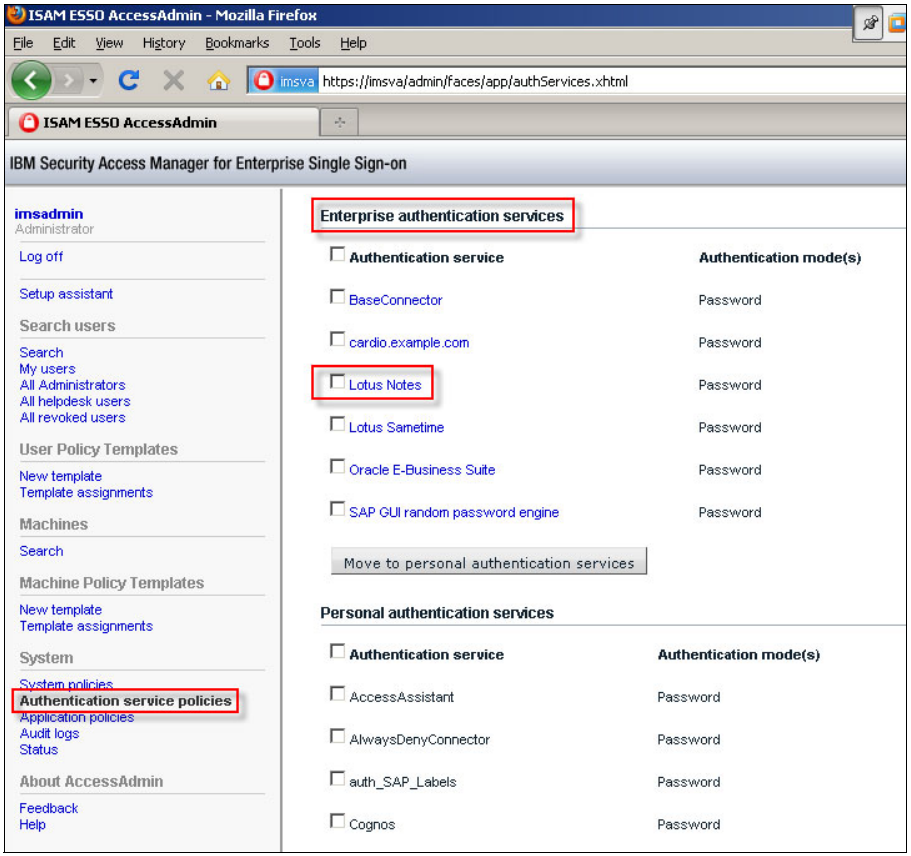

Figure 5-78 Authentication service policies

3. Ensure that the default sign-on action for Lotus Notes is set to Automatic Logon and that it is set to close the application when the user logs off AccessAgent. Under System, click **Application policies**. Select **Lotus Notes** (Figure 5-79).

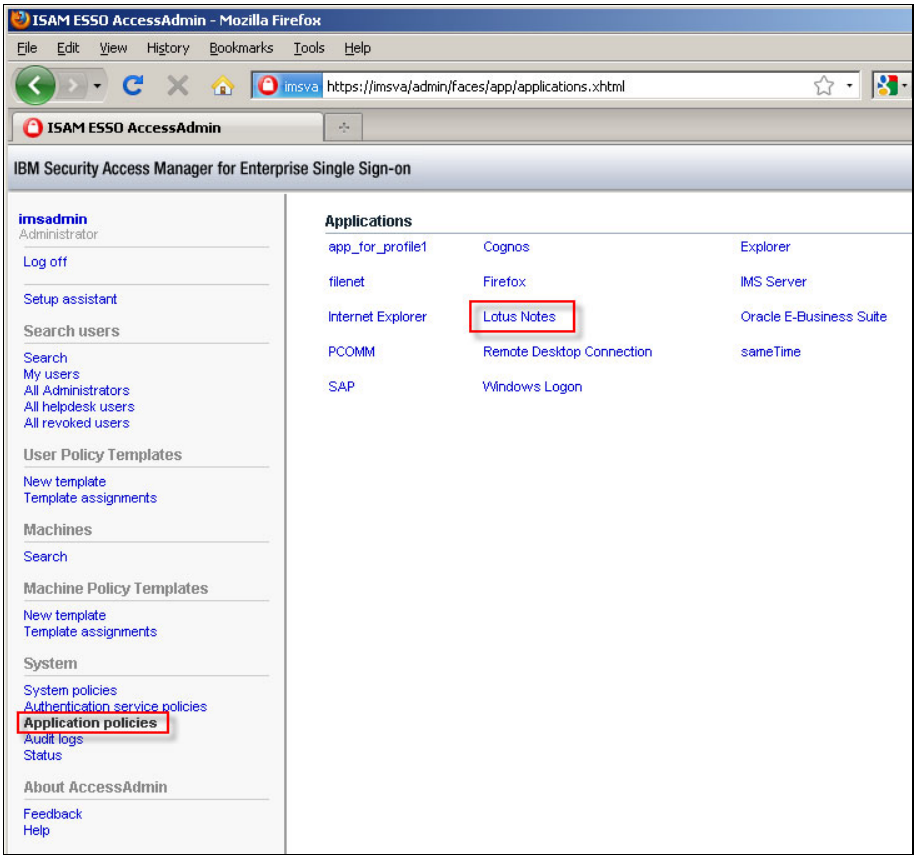

Figure 5-79 In Application policies, select Lotus Notes

4. In the drop-down (Default single sign-on using the automatic sign-on mode password entry option for the application), select **Automatic logon** (Figure 5-80).

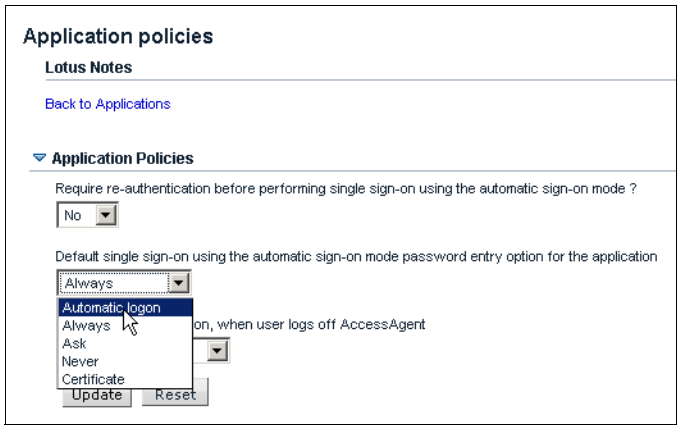

Figure 5-80 Set Automatic logon as the default action for all users

5. In the drop-down (Action for the application, when user logs off AccessAgent), select **Close the application** (Figure 5-81).

Figure 5-81 When a user logs off AccessAgent, Lotus Notes needs to close

6. Click **Update** to apply the changes.

After we change these options, the IBM Security Access Manager for Enterprise Single Sign-On AccessAgents must synchronize with the IMS Server to receive

the changes. Then, AccessAgents act when a user starts and logs in to Lotus Notes. The next steps illustrate this process:

1. After the user logs in to Windows and starts Lotus Notes, the password dialog box opens as usual (Figure 5-82).

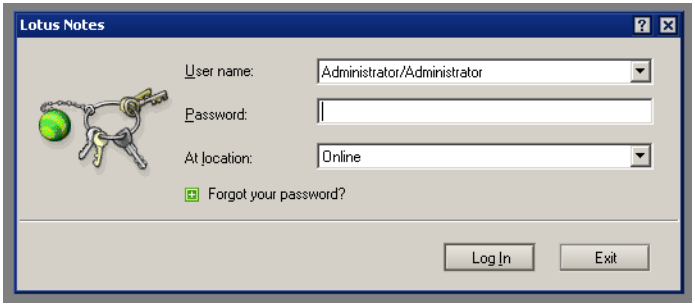

Figure 5-82 Lotus Notes Password prompt

2. To show that IBM Security Access Manager for Enterprise Single Sign-On does not attempt to save incorrect credentials, the user enters an incorrect password (Figure 5-83). Because the manually entered password is incorrect, AccessAgent takes no action.

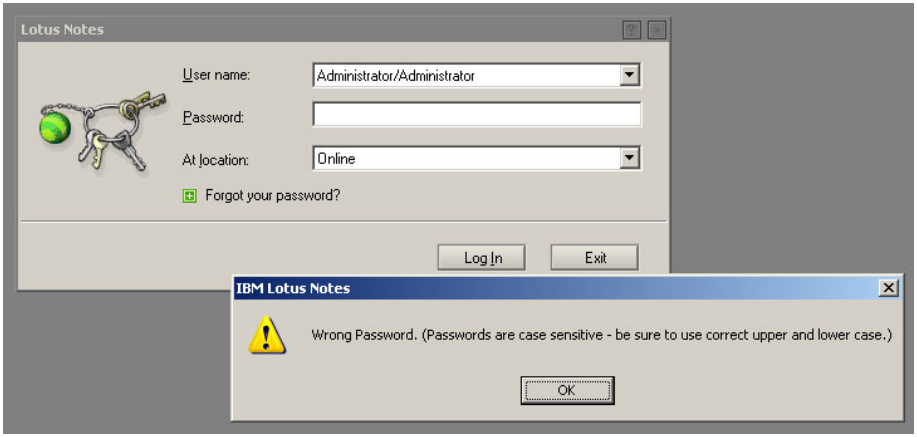

Figure 5-83 Due to a manually entered incorrect password, no action is taken

Important: The default AccessProfile for Lotus Notes is already configured to detect when an incorrect password is entered. When creating new AccessProfiles, this functionality must be configured manually.

3. The user now enters the correct password, AccessAgent detects it, and AccessAgent prompts the user to save the credentials in the Wallet (Figure 5-84).

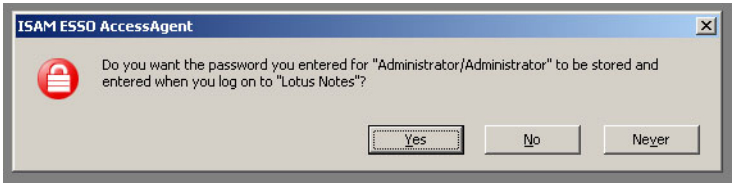

Figure 5-84 AccessAgent prompts user to save user's Lotus Notes credentials

4. The user is now authenticated and logged in to Lotus Notes (Figure 5-85).

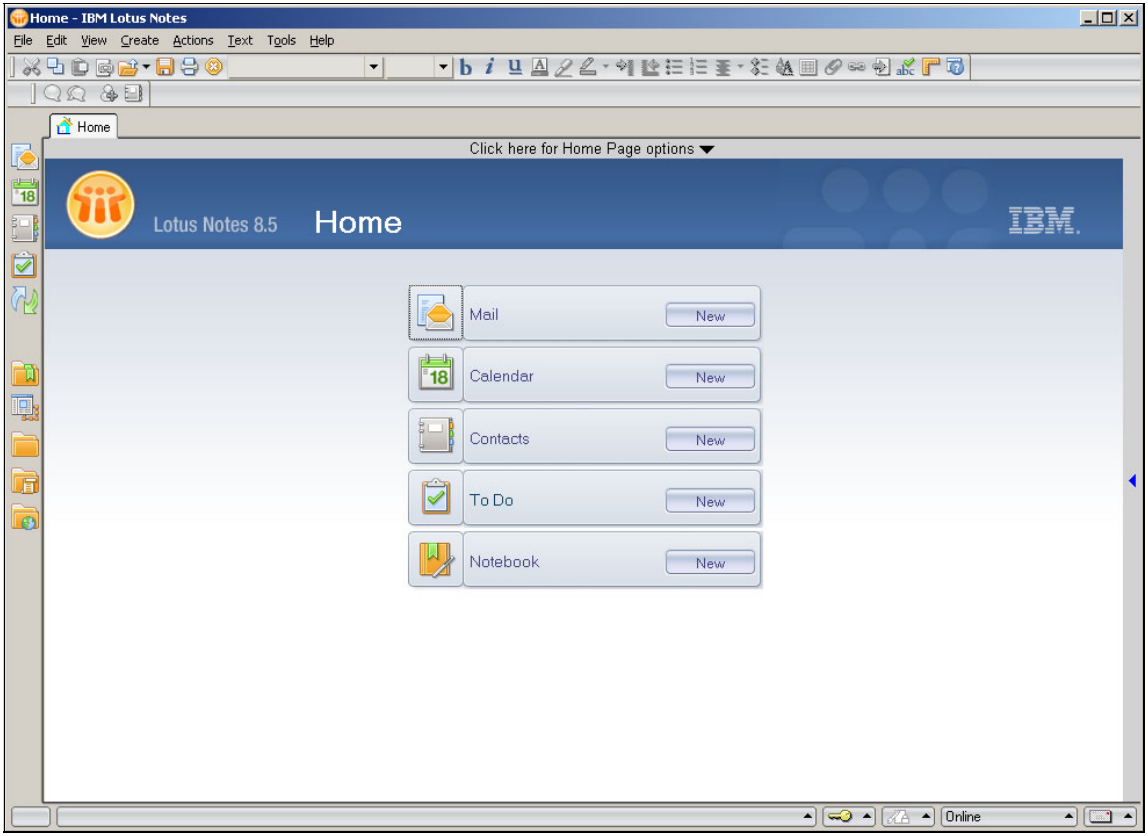

Figure 5-85 User is authenticated and logged in to Lotus Notes

5. To see that the credentials are saved to the Wallet, right-click the AccessAgent icon in the system tray, and select **Manage Wallet** (Figure 5-86).

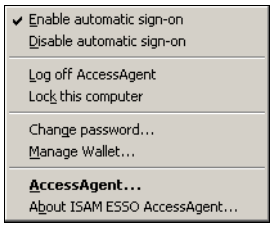

Figure 5-86 Manage Wallet

6. The user now has Lotus Notes credentials saved in the Wallet. It is an enterprise authentication service, and the default logon action is set to Automatic Logon (Figure 5-87).

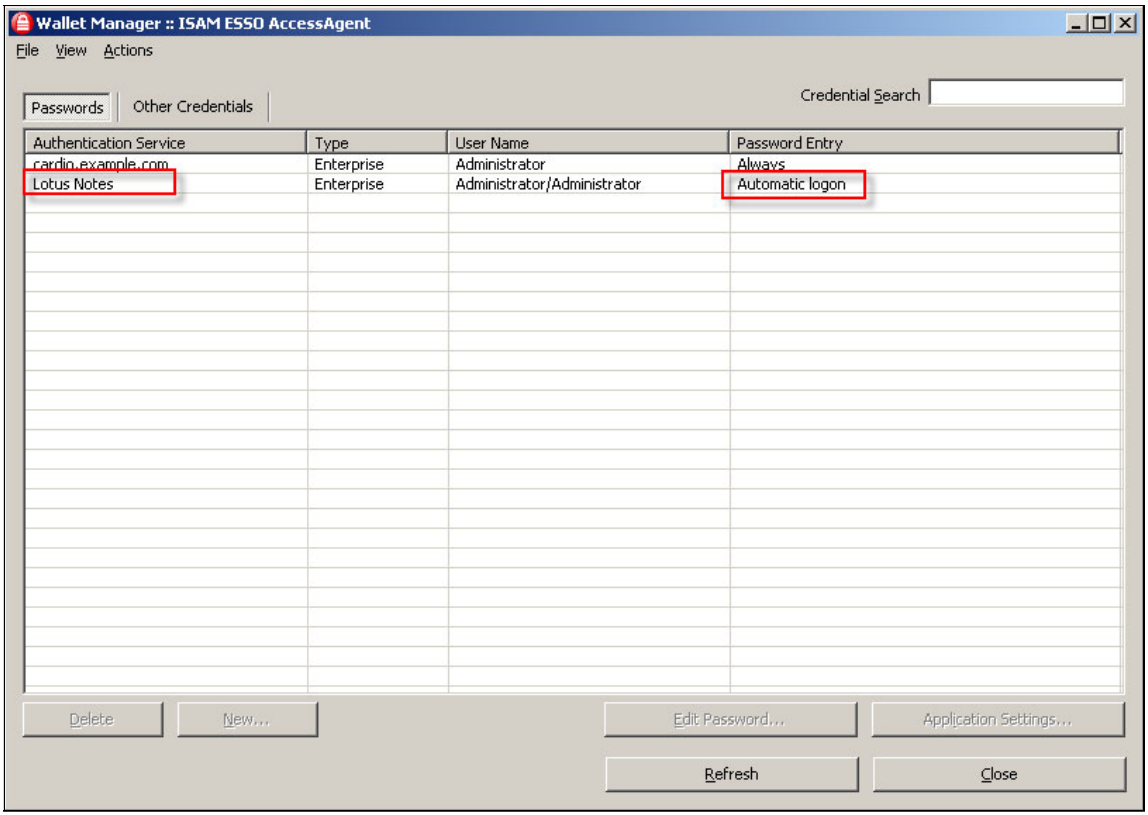

Figure 5-87 Wallet contains the Lotus Notes authentication service and credentials

From this point forward, when the user opens Lotus Notes, IBM Security Access Manager for Enterprise Single Sign-On automatically injects and submits the user credentials.

If the credentials are not submitted automatically: If IBM Security Access Manager for Enterprise Single Sign-On does not submit the credentials automatically, you might either check for different profiles online or adjust the profile itself. Navigate to the action in AccessStudio that submits the credentials and change it to automatically submit the credentials.

The Lotus Notes configuration is complete. Next, we examine the SAP application.

5.3.2 SAP application

The cardio healthcare company relies on SAP applications to maintain patient records and to administer the billing process. Most of the employees need to interact with various SAP platforms on a daily basis. This interaction is a cause for concern, because the SAP applications all have complex password requirements, and the account credentials are not synchronized across the SAP systems. The SAP application is one of the primary candidates for the single sign-on project.

In the following sections, we examine these details:

- Single sign-on considerations
- System name display requirements
- Implementation process overview
- How the AccessProfile works
- Single sign-on considerations

Single sign-on considerations

With any application, it is essential to understand the requirements for that application at the outset. Consider the following factors when planning to use IBM Security Access Manager for Enterprise Single Sign-On to provide single sign-on to SAP applications:

- How many different SAP systems are used, and what are the names of those systems?
- What languages are used to access the SAP systems?
- How do users access the SAP systems? Do they use the SAP Windows client, a web browser, or both?

- If the users use the SAP Windows client, what versions of the client do they use?
- Are SAP application shortcuts used?
- If the users use a web browser, what URLs are used to access the SAP systems?
- How many fields do users enter when logging on to SAP applications? Is it just the user name and password, or does the user also need to enter the client name and language?
- How do we want to display the names of the SAP systems in the IBM Security Access Manager for Enterprise Single Sign-On Wallet manager?
- Is there any synchronization of the passwords between different SAP systems? (That is, if the user changes the password on one SAP system, does this change affect on any other SAP system?)

When determining the answers to these questions, ensure that you consider all likely groups of users. The answers to these questions affect how to best implement IBM Security Access Manager for Enterprise Single Sign-On.

System name display requirements

When the SAP Windows client is started, a list of systems is displayed. The system names that are initially displayed are the values of the Description fields for the various systems.

Where are my attributes: The Description, Server, Database, System, Attribute, and other attributes for the various systems are stored in the `saplogon.ini` file that is installed with the SAP Windows client on the Windows system.

The user might select, for example, a system whose description is `A1 : ABCD - Production`. The SAP Windows client then establishes a connection with the appropriate system based on information from `saplogon.ini`.

Alternatively, the user might access the same SAP system through a web browser by specifying this URL:

```
https://sysabcd.cardio.example.com/
```

You must decide how to display the system names in the IBM Security Access Manager for Enterprise Single Sign-On Wallet manager (the *authentication service* name). It is important that you focus on this requirement to determine the best way to display these names. It is especially important if both web browsers and SAP Windows clients are used by the same users to access the same

systems, particularly where the respective names bear little resemblance to each other.

We suggest these approaches:

- Where the SAP system is accessed solely through the web browser, the authentication service name can correspond to the fully qualified domain name of the SAP system URL (such as `sysabcd.cardio.example.com`).
- Where the SAP system is accessed solely through the SAP Windows client, the authentication service name can correspond to the description field from `saplogon.ini` (such as `A1 : ABCD - Production`).
- Where the SAP system is accessed either through the web browser or through the SAP Windows client, the authentication service name must include both the description and the URL (such as `A1 : ABCD - Production - sysabcd.cardio.example.com`).

Next, we describe how to meet these display name requirements.

Implementation process overview

We provide an overview of the steps involved in implementing IBM Security Access Manager for Enterprise Single Sign-On for SAP applications:

1. Determine the SAP application single sign-on considerations.
2. The standard IBM Security Access Manager for Enterprise Single Sign-On AccessProfile for SAP applications can be obtained by downloading the latest IBM Security Access Manager for Enterprise Single Sign-On Software Profile Bundle from the IBM support site (component ID 5724N70IF)[2].
3. Deploy the IBM Security Access Manager for Enterprise Single Sign-On AccessProfile for SAP applications to the IMS Server.
4. Make the necessary client workstation configuration changes (as described in the release notes for the IBM Security Access Manager for Enterprise Single Sign-On SAP profile, which is included in the bundle).
5. Create the required authentication services.
6. Test and tailor the AccessProfiles, as required.

How the AccessProfile works

In this section, we describe the IBM Security Access Manager for Enterprise Single Sign-On application profile.

[2] The direct link to this resource is https://www.ibm.com/support/docview.wss?uid=swg24029132.

Application process control

The SAP GUI window is the window where SAP logon credentials are entered, and therefore, the window from which the AccessProfile must capture and inject credentials. Without IBM Security Access Manager for Enterprise Single Sign-On deployed on an SAP client system, the SAP GUI window opens as part of the SAP Logon executable process and not as a separate application or process. Only one instance of the SAP Logon application process exists, but multiple SAP GUI windows can exist. IBM Security Access Manager for Enterprise Single Sign-On AccessProfiles are initiated when executable processes are started, and not when windows are rendered. So, an AccessProfile cannot cater to a process model where an unknown number of windows exist with each window that requires credential capture and injection.

To handle this process model, the SAP AccessProfile ensures that each SAP GUI window is owned by a separate application process. The SAP Logon application process destroys the SAP GUI window when it opens. The SAP Logon application process then creates an instance of the SAP GUI application process and therefore a new instance of the SAP GUI window.

The SAP AccessProfile consists of two parts. The first part, profile_SAP_LOGON, is associated with the `saplogon.exe` application. The second part, profile_SAP_GUI, is associated with the `sapgui.exe` application.

SAP client system configuration information is shared between the two parts of the SAP AccessProfile. This configuration is based on the SAP configuration that is stored in the `saplogon.ini` file, which is read by the profile_SAP_LOGON part of the SAP AccessProfile.

Number of capture or injection fields

Typically, four entry fields appear on the SAP GUI logon window:

- Client field
- User name field
- Password field
- Language field

The standard release of the AccessProfile for SAP supports the capture of all four window fields. However, the client and language fields are optional. The user name and password credentials are still captured even if those fields are left blank by the user. Other custom AccessProfiles for SAP can support fewer than four fields, such as the user name and password fields only.

Single sign-on to SAP applications by using SAP Logon

Two SAP profiles are supported with IBM Security Access Manager for Enterprise Single Sign-On.

We look closely at these SAP profiles:

- SAP Logon AccessProfile (profile_SAP_LOGON)

  The profile_SAP_LOGON is instantiated when the `saplogon.exe` application process is started. The profile_SAP_LOGON remains in the state_setup state until the SAP Logon application main window opens (Figure 5-88).

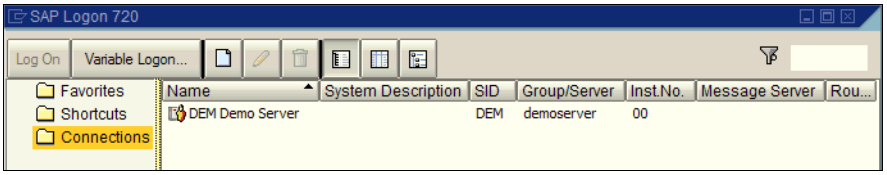

Figure 5-88 SAP Logon Main Window (source: SAP AG)

  When the window opens, the profile_SAP_LOGON transitions to the state_start_sapgui state. During that transition, the profile parses the `saplogon.ini` file for SAP server connection configuration information and stores that information in a global array.

  The profile_SAP_LOGON waits for you to click **Log On** or the appearance of the main SAP GUI logon window, at which point the profile_SAP_LOGON transitions to the state_close_launched_window state.

  The profile_SAP_LOGON profile then destroys the main SAP GUI logon window, because that window is not associated with a separate application process. The destruction of that window occurs through the Close A Window action on all triggers, leading to a transition to the state_relaunch state. After a short delay, the profile_SAP_LOGON initiates a script that invokes the SAP GUI application process as a separate instance during the state transition to the state_launched state. In turn, the profile_SAP_GUI is instantiated for that instance of the SAP GUI application process.

- SAP GUI AccessProfile (profile_SAP_GUI)

  The profile_SAP_GUI is instantiated when the `sapgui.exe` (or a similar) application process is started. The profile_SAP_GUI remains in the state_setup state until the SAP GUI main window opens.

  When that window opens, the profile_SAP_GUI transitions to the SAP_GUI_Logon_Window_Displayed state. During that state transition, a script is run to copy the globally stored SAP configuration information into local storage.

  Immediately, a transition to the Was_Stored_Cred_Injected state occurs, during which an injection of the first screen entity is attempted after first fetching credentials from the Wallet. The particular entity injected depends on

the number of screen fields supported by the profile (see “Number of capture or injection fields” on page 179).

Depending on whether the injection was successful, the built-in `NO_ACCOUNT_DATA_FOUND` flag is set or cleared. The profile flow follows one of two paths, which join again at the Creds_Injected state. The following state transition path is followed if credentials were found in the Wallet: **Was_Stored_Cred_Injected** → **Stored_User_Injected** → **Stored_Password_Injected** → **Stored_Language_Injected** → **Creds_Injected**.

If no credentials were found in the Wallet, the state transition path is **Was_Stored_Cred_Injected** → **Was_Capture_Cancelled** → **Captured_Client_Injected** → **Captured_User_Injected** → **Captured_Password_Injected** → **Captured_Language_Injected** → **Creds_Injected**.

The profile_SAP_GUI uses the *Show a dialog to capture logon credentials* action to capture the required entry fields (Figure 5-89).

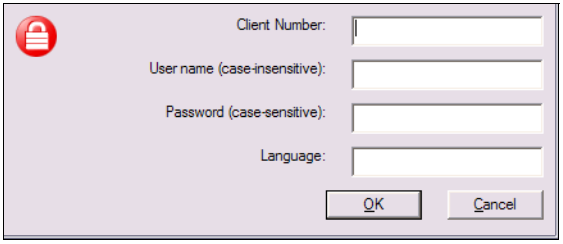

Figure 5-89 Capture Logon Credentials dialog (source: SAP AG)

From the Creds_Injected state, the profile handles the possible error messages that can occur during the logon attempt with the user-entered screen entry field contents.

Several other windows might open when in this state. First, the Information window can open. When this window opens, the profile waits for the information window to be cancelled before proceeding.

The other window that can open is the password change dialog, which is started by using clicking **New Password** or when the existing password expires. When this dialog opens, the profile uses the *Show a dialog to capture change password credentials* action to capture the new password (Figure 5-90 on page 182). The profile caters for errors that occur during the password change.

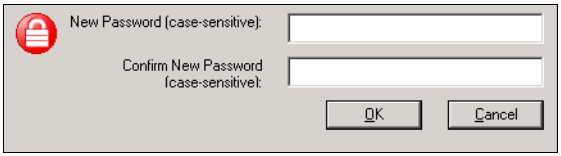

Figure 5-90 Change password dialog

Successful logon to the selected SAP system occurs when the main SAP GUI window opens. This window is called the SAP Easy Access window. When this window opens, the profile transitions from the Creds_Injected state to the LoggedOn state. This profile never returns to the start_state state.

Single Sign-On to SAP applications by using the SAP Shortcut

In this section, we look at three use cases:

- Launching Shortcut from SAP Logon (profile_SAP_LOGON)

  The profile_SAP_LOGON is instantiated when the `saplogon.exe` application process is started. The profile_SAP_LOGON remains in the state_setup state until the SAP Logon application main window (Figure 5-88 on page 180) opens. When the window opens, the profile waits for the user to launch a shortcut from the Shortcuts tab. After a shortcut is launched, the profile transitions to the state_after_logon_sap_shortcut state, during which it executes a script that reads information from the `sapshortcut.ini` file. The profile then transitions back to the state_launched state. It closes the SAP Shortcut window while launching a new separate process that initiates the shortcut so that the profile_SAP_GUI profile can be instantiated.

- Launching Shortcut from the desktop (profile_SAP_LOGON)

  The profile_SAP_LOGON AccessProfile is instantiated when an SAP Shortcut is launched from the desktop. When the profile detects that a shortcut is launched, it transitions to the state_initial_SAP_Shortcut state. The profile closes the shortcut window while launching it in a separate process, similar to the method described in “Single sign-on to SAP applications by using SAP Logon” on page 179. The profile then transitions to the state_launched state to wait for any user action with the SAP Logon application (that is, connect to other SAP Systems) and handles the user action as described in the previous sections.

- SAP Shortcut in SAP GUI (profile_SAP_GUI)

  When the profile_SAP_GUI AccessProfile is instantiated and detects that an SAP Shortcut window is launched, it transitions to the state SAP_Shortcut_Window_Displayed. A script is then executed to determine the name of the authentication service to use. Eventually, after injecting and capturing credentials, the profile transitions into the Creds_Injected state. The

profile is handled the same way that is described in “SAP GUI AccessProfile (profile_SAP_GUI)” on page 180.

AccessProfile language support

The AccessProfile for SAP applications supports multiple languages, and each language is handled by the profile. To add additional language support, the profile must be modified. Specifically, the login error messages are language-specific. For example, if an incorrect user name and password are entered, the English language error text displayed is `Name or password is incorrect (repeat logon)`. In German, it is `Name oder Kennwort ist nicht korrekt (Wiederholen Sie die Anmeldung)`.

There are equivalent error messages for other languages that the SAP system supports. The profile_SAP_GUI uses regular expressions to abbreviate the text required to be entered to these error messages. In the case of the previous example, this example is the entire regular expression string:

```
(.*incorrect.*)|(.*nicht korrekt.*)|(.*niet
correct.*)|(.*incorrectos.*)|(.*errati.*)|(.*correto.*)
```

This expression string handles the error message in English, German, Dutch, Spanish, Italian, and Portuguese.

Appropriate regular expression strings are used for other error messages.

AccessProfile environment and default settings

There are four key SAP client system environment settings that affect the AccessProfile for SAP applications. These settings are in the script in the state_start to state_setup state transition part of the profiles (Figure 5-91 on page 184):

- Location of the `saplogon.ini` file
- Name and location of the SAP GUI application executable file (`SAPGui.exe`)
- Location of `sapshortcut.ini` file
- Name and location of the SAP Shortcut application executable file (`sapshcut.exe`)

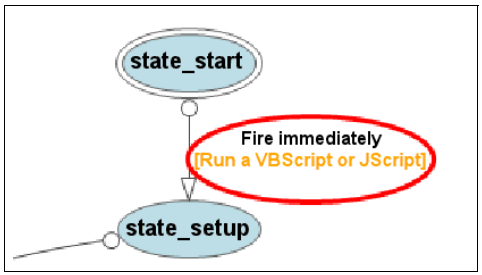

Figure 5-91 State transition from state_start to state_setup state in AccessStudio

In the context of the AccessProfile environment and default settings, we first look at the default location of system environment settings and then examine how to edit the AccessProfile to customize system environment settings:

- Default location of system environment settings

  The profile assumes that all settings are in their default locations. If they are not in their default locations, the profile must be modified to use the location. These locations are the default locations:

  - `saplogon.ini` is in one of the following paths:
    - Environment variable `SAP_LOGON_INI`
    - `%APPDATA%\SAP\Common\saplogon.ini`
    - Registry key at `HKLM\SOFTWARE\SAP\SAP Shared\SAPSysdir`
    - `%SYSTEMROOT%` directory

    For example:

    `C:\Users\AppData\SAP\Common\saplogon.ini`

  - SAP GUI application executable file (`SAPGui.exe`) is expected at this location:
    - Registry key at `HKLM\SOFTWARE\SAP\SAP Shared\SAPSysdir`
    - `%PROGRAMFILES%\SAP\FrontEnd\SAPgui\` directory

    For example:

    `C:\Program Files\SAP\FrontEnd\SAPgui\SAPGui.exe`

  - `sapshortcut.ini` is expected to be in the same folder as `saplogon.ini`:
    - Environment variable `SAP_LOGON_INI`
    - `%APPDATA%\SAP\Common\saplogon.ini`
    - Registry key at `HKLM\SOFTWARE\SAP\SAP Shared\SAPSysdir`
    - `%SYSTEMROOT%` directory

    For example:

    `C:\Users\AppData\SAP\Common\sapshortcut.ini`

– SAP Shortcut Launcher application executable file (`SAPshcut.exe`) is expected at this location:

  - Registry key at `HKLM\SOFTWARE\SAP\SAP Shared\SAPSysdir`
  - `%PROGRAMFILES%\SAP\FrontEnd\SAPgui\` directory

  For example:

  `C:\Program Files\SAP\FrontEnd\SAPgui\SAPshcut.exe`

► Editing the AccessProfile to customize system environment settings

In certain client systems, these environment settings can differ from the default. In this case, the system environment settings can be customized by editing the AccessProfiles in AccessStudio:

– In AccessStudio, edit the VBScript code in the profile_SAP_LOGON when the state transitions from state_start to state_setup (Figure 5-92).

– Similarly, in AccessStudio, edit the VBScript code in the profile_SAP_GUI when the state transitions from state_start to state_setup.

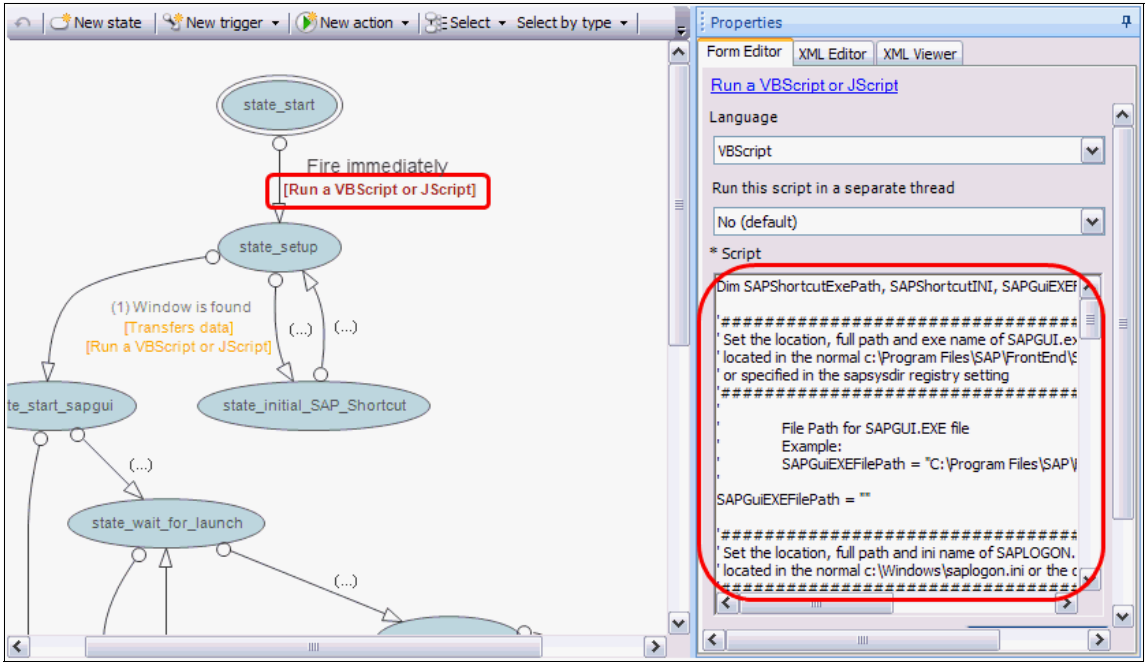

Figure 5-92 Editing the script in between state_start to state_setup

Setting up the registry for SAP Logon

To enable the AccessProfile to work correctly (as described in the “Application process control” on page 179), a registry setting for SAP applications must be modified. There are three methods to perform this task. The first method is by using the AccessAgent pre-installation task, where the registry key is set when installing AccessAgent on the client machine. The second method is by manually editing the registry key. The third method is by using the AccessProfile to automatically set it when SAP Logon is launched.

We define the methods in more detail:

- Method 1: Use AccessAgent pre-installation task. Follow these steps:
  a. Edit the `Reg\DeploymentOptions.reg` file in the AccessAgent installation directory.
  b. Add the following lines to the file:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\SAP\SAPGUI]
"StartSapLogon"=dword:00000000
```

- Method 2: Manually edit the registry. Follow these steps:
  a. Launch Registry Editor:

     For Windows 7:

     i. Click the **Start** menu icon in the lower-left corner of your desktop.
     ii. Click the search field above the Start menu button and type `regedit` into it.
     iii. Click **Regedit** in the search results to launch the Registry Editor.

     For Windows XP and Windows Vista:

     i. Click the **Start** menu button, then click **Run**.
     ii. Type `regedit` into the Open text box.
     iii. Click **OK** in the Run window to launch the Registry Editor.

  b. Navigate to the key `HKEY_LOCAL_MACHINE\SOFTWARE\SAP\SAPGUI`.
  c. Create a DWORD value StartSapLogon (if it does not exist).
  d. Edit the DWORD value StartSapLogon and set it to `0`.
  e. Close the Registry Editor.
- Method 3: AccessProfile automatic setting on first launch. Follow these steps:
  a. Load the AccessProfile profile_SAP_LOGON in AccessStudio.
  b. Edit the VBScript that is executed in the state transition from state_setup to state_start_sapgui (Figure 5-93 on page 187).

c. Change the line in the beginning of the script from `isAutomaticRegistrySetting = 0` to `isAutomaticRegistrySetting = 1`.

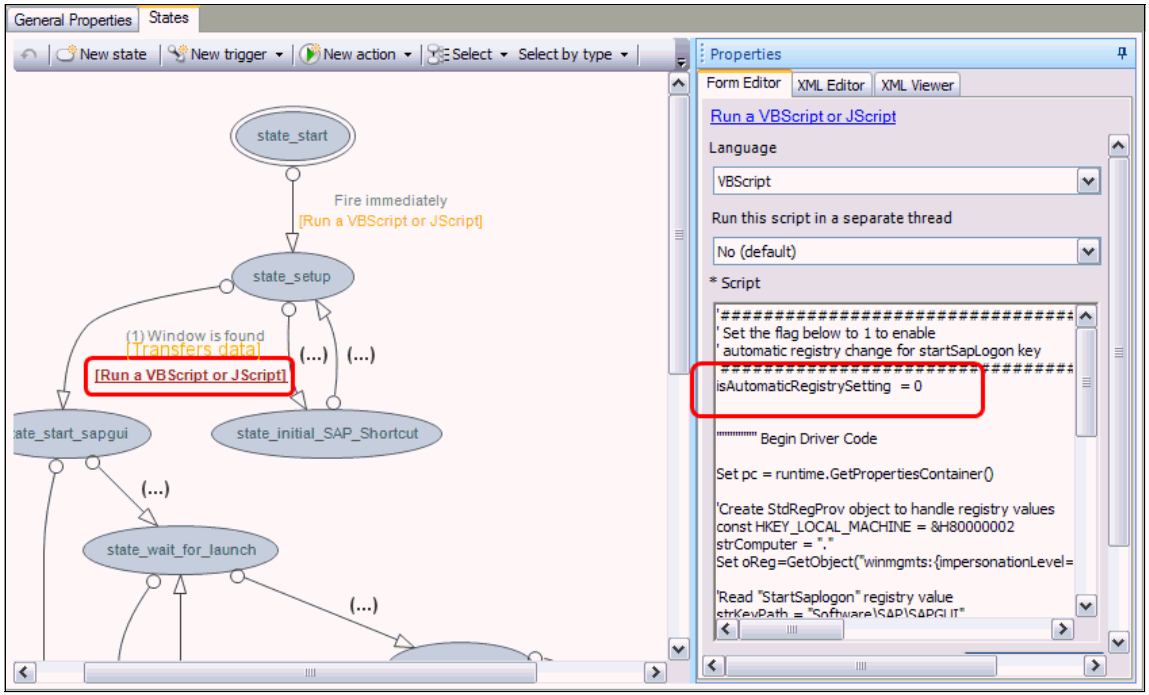

Figure 5-93 AccessStudio editing the script in between state_setup to state_start_sapgui

Capturing credentials when signing in to SAP applications

To capture credentials when first signing in to SAP applications, follow these steps:

1. Start the SAP Logon application and select the appropriate system on which to sign on.
2. When the SAP GUI application loads, wait for the IBM Security Access Manager for Enterprise Single Sign-On credentials capture-prompt to appear (Figure 5-89 on page 181).
3. Enter the credentials into the appropriate text fields. Click **OK**.
4. IBM Security Access Manager for Enterprise Single Sign-On prompts you to save the credentials to your Wallet. Click **Yes** to save.

The user is now logged in to the SAP application and the credentials are now saved for subsequent single sign-on to this system.

Generating random password during password change

The AccessProfile for SAP GUI includes the capability to generate a random password, based on your password policy set in the IMS Server. The AccessProfile has an enterprise authentication service auth_SAP_random_password_engine that is uploaded to the IMS Server together with the AccessProfile. Administrators can set the random password generator policy for SAP GUI credentials in the IMS Server by selecting **AccessAdmin → System → Authentication service policies → SAP GUI random password engine**.

Only the password policies are applicable (Figure 5-94). All other authentication policies in the auth_SAP_random_password_engine authentication service do not apply to the SAP GUI AccessProfile.

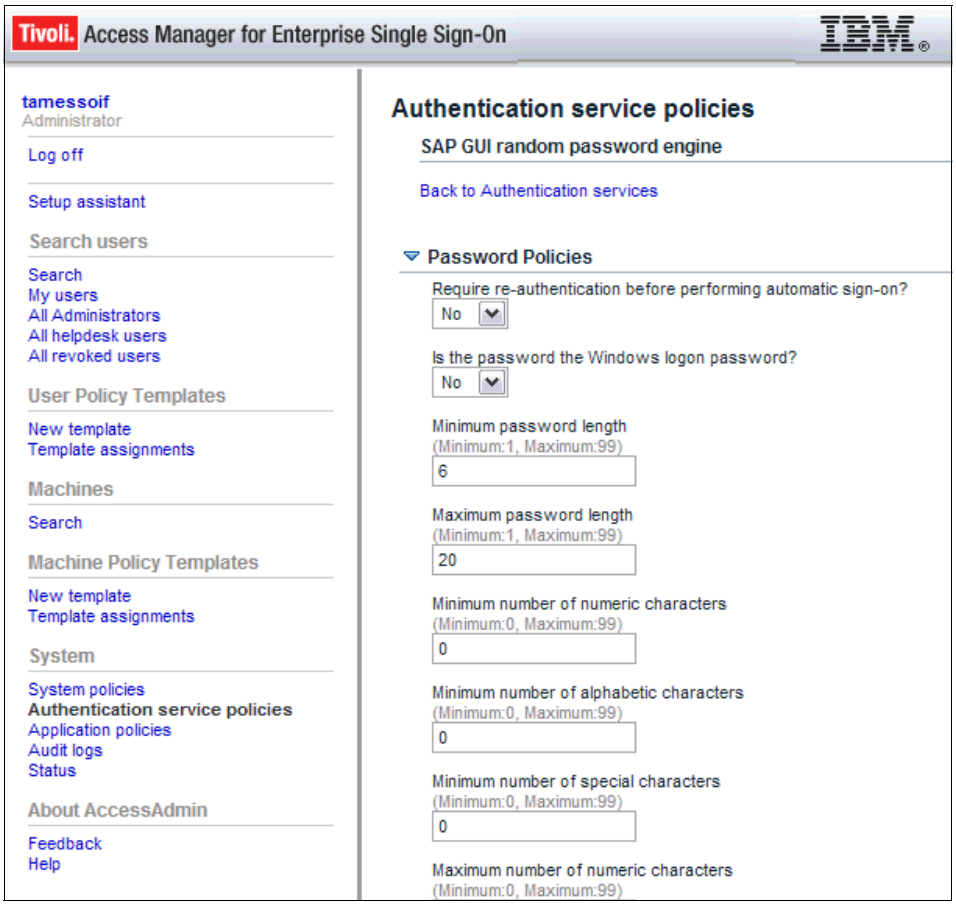

Figure 5-94 SAP GUI random password engine authentication service policies

How does that work? The AccessProfile runs a JScript when the SAP GUI shows the change password dialog box. This JScript downloads the password policy from the auth_SAP_random_password_engine authentication service in the IMS Server and then generates a new random password based on the policy. It then injects the new random password into the Change password dialog box.

With the current design, users can change (that is, delete and re-enter) a new password of their choice. If this behavior is undesirable, it is easy to add an action to click **OK** after the new password is injected. The Click action button must be added in the profile_SAP_GUI state transition from AA_ChangePwd_Shown to Wait_AAChangePwd_Destroyed.

Authentication service names

The *authentication service name* is the name that users see in their Wallet. The ideal configuration depends on whether the SAP applications are accessed either through a web browser, the SAP Windows client, or both. Let us examine the different access options:

- Access via web browser only

  Create a validating web browser AccessProfile for single sign-on to the SAP web interface. That is, use the AccessStudio AccessProfile Generator to create an AccessProfile where the window that appears on successful logon is identified. This method avoids capturing incorrectly typed credentials.

  By default, the authentication service name displayed in the IBM Security Access Manager for Enterprise Single Sign-On Wallet manager is the host name that is used to specify the SAP system (such as `sysabcd.cardio.example.com`).

  If you want a different display name, follow these steps:

  a. Use the IBM Security Access Manager for Enterprise Single Sign-On AccessStudio to create an authentication service with the display name that you want.

  b. Create a server locator for that authentication service that matches the host name that is used to specify the SAP system (such as `sysabcd.cardio.example.com`).

- Access via the Windows SAP client only

  By default, the authentication service name displayed in the IBM Security Access Manager for Enterprise Single Sign-On Wallet manager is the server name specified in the `saplogon.ini` file for the specified SAP system.

  If you want an authentication service name displayed in the IBM Security Access Manager for Enterprise Single Sign-On Wallet manager that matches the description field from `saplogon.ini` (such as `A1 : ABCD - Production` from the previous example), edit the AccessProfile according to the

instructions in “Change the authentication service naming convention in AccessProfile” on page 190.

- Access via both web browser and Windows SAP client

  By default, the authentication service names displayed in the IBM Security Access Manager for Enterprise Single Sign-On Wallet manager are the names we described previously, depending on whether a web browser or the SAP client is used to access the SAP system. (It is undesirable to have two different entries in the Wallet for the same SAP system.)

  If you want a display name that includes both the description and the URL (such as `A1 : ABCD - Production - sysabcd.cardio.example.com`), take the following steps:

  a. Use IBM Security Access Manager for Enterprise Single Sign-On AccessStudio to create an authentication service with the desired display name (such as `A1 : ABCD - Production - sysabcd.cardio.example.com`).

  b. Create a server locator for that authentication service that matches the web browser host name used to specify the SAP system (such as `sysabcd.cardio.example.com` from the previous example).

  c. Create a server locator for that authentication service that matches the authentication service name generated by the AccessProfile when the Windows SAP client is used (such as a `sapsys2.cardio.example.com` server name).

  d. Alternatively, in place of the previous step, modify the AccessProfile as described in the previous section to create a server locator for that authentication service that matches the description from `saplogon.ini` (such as `A1 : ABCD - Production`).

- Change the authentication service naming convention in AccessProfile

  The default naming convention used by the AccessProfile is in the following form:

  ```
  [Description]_[Application Server / Message Server / Router]
  ```

  You can change the naming convention used by the AccessProfile, if required. In profile_SAP_Logon AccessProfile, edit the script that is executed between state_relaunch state and state_wait_for_launch_again state. At the beginning of the script, locate the following line:

  ```
  strAuthService = global_profile_sap_logon_desc & "_" &
  strAuthService
  ```

  The statements to the right side of the equals sign (=) can be replaced with any combination of the variable names in Table 5-1 on page 191. The resulting statement must be a valid VBScript syntax.

You must duplicate the changes to the script that is executed between the state_setup state and SAP_Shortcut_Window_Displayed state found in profile_SAP_GUI AccessProfile.

Table 5-1 Available variables for use with authentication service ID naming

Variable name	Description
global_profile_sap_logon_server	The application server
global_profile_sap_logon_router	Router to the application and message server
global_profile_sap_logon_mssysname	Three-character abbreviation of SAP system ID
global_profile_sap_logon_database	SAP database number
global_profile_sap_logon_mssrvname	Host name of the message server
global_profile_sap_logon_mssrvport	The port number of the TCP link
global_profile_sap_logon_desc	Description of an SAP link as it appears in the SAP Logon
global_profile_sap_logon_origin	Type of entry: MS_SEL_GROUPS, MS_SEL_SERVER, or USEREDIT

Web single sign-on to SAP GUI for HTML

For many SAP system users, the SAP GUI application is the mechanism used to access their SAP systems. However, certain users in the cardio healthcare company use the SAP GUI for HTML (SAP WebGUI) interface instead.

Single sign-on to the SAP GUI for HTML is also possible with IBM Security Access Manager for Enterprise Single Sign-On by using the *Basic Authentication* access control mechanism. IBM Security Access Manager for Enterprise Single Sign-On supports *Basic Authentication* with two browser profiles: one browser profile for Internet Explorer and one browser profile for Mozilla Firefox. These profiles are general web page *Basic Authentication* credential capture and inject profiles, and they are not specific to SAP systems.

SAP GUI for HTML

The SAP GUI for the HTML URL is in this form:

```
http://10.150.22.2:8000/sap/bc/gui/sap/its/webgui
```

Figure 5-95 on page 192 shows the logon web page.

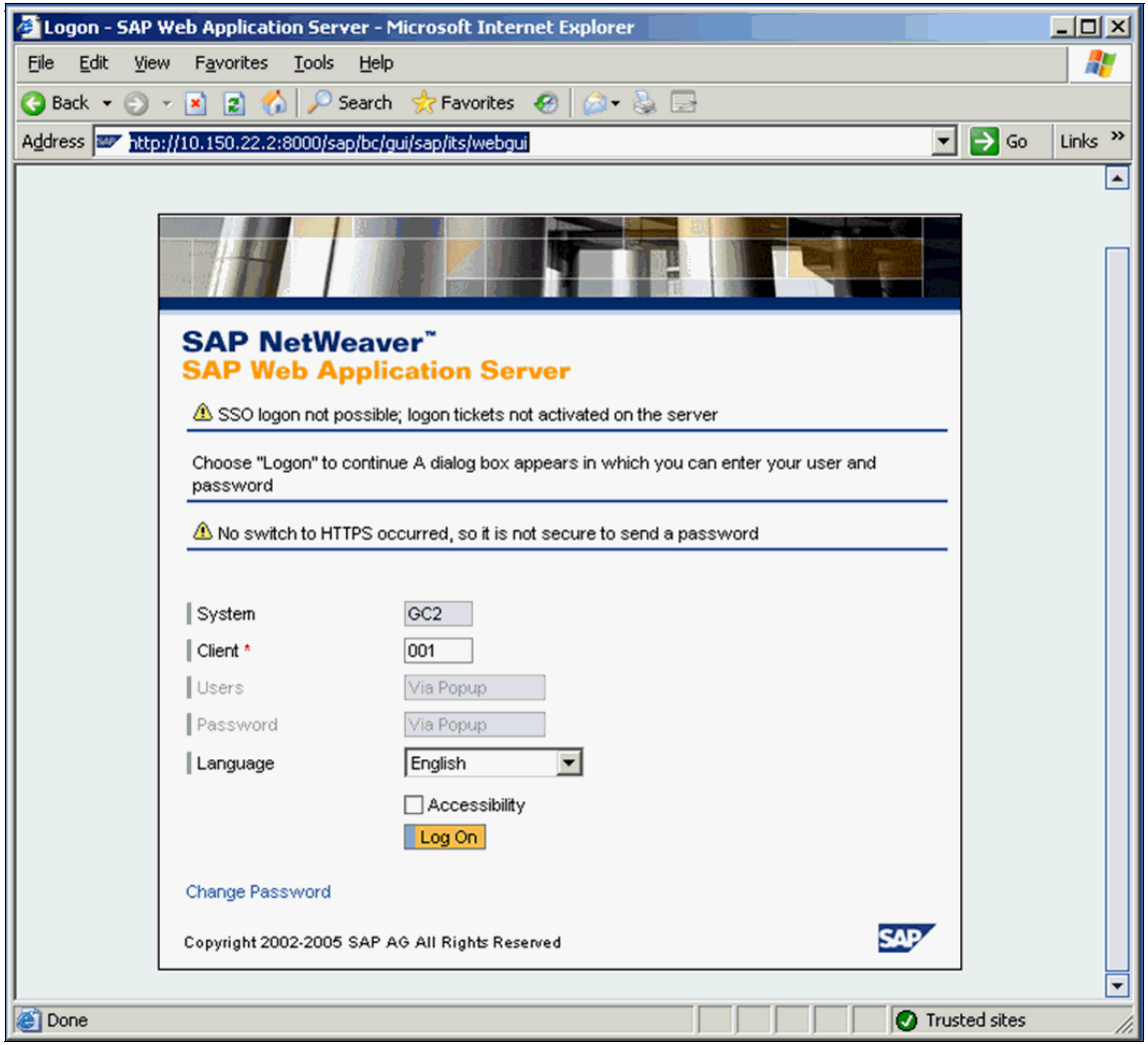

Figure 5-95 SAP WebGUI logon web page (source: SAP AG)

Clicking **Log On** opens the Basic Authentication prompt dialog.

Web Single Sign-On by using Internet Explorer browser

Figure 5-96 shows the Basic Authentication prompt dialog for Internet Explorer.

Figure 5-96 SAP WebGUI Basic Authentication with Microsoft Internet Explorer

The IBM Security Access Manager for Enterprise Single Sign-On profile for Internet Explorer Basic Authentication is called `sso_site_wnd_iexplore` and has these characteristics:

- Injecting SAP logon credentials by using the Internet Explorer profile

  From the state_start state, the profile transitions to state_after_inject via a *Window is activated* trigger, which triggers when the Internet Explorer Basic Authentication prompt dialog is displayed. During this state transition, stored credentials are injected into the Basic Authentication prompt dialog. There are multiple instances of this trigger in the Internet Explorer profile to handle the Basic Authentication prompt dialog presented in different languages.

  The following line is the Windows XPath signature for the English language version of this trigger:

  ```
  /child::wnd[@title~"Connect to(.*)"]/child::wnd[@class_name="Button"
  and @ctrl_id=1]/parent::wnd[@class_name="#32770"]
  ```

  Click **OK** to submit the injected credentials to the SAP application for authentication. There is no validation of the correct logon with the injected credentials.

- Capturing SAP logon credentials by using the Internet Explorer profile

  When you click **OK** in the Basic Authentication prompt dialog, credentials entered in the Basic Authentication prompt dialog are stored in the IBM Security Access Manager for Enterprise Single Sign-On Wallet. There is no validation of the correct logon with the injected credentials.

The state transition that occurs when you click **OK** is state_after_inject to state_after_inject.

Authentication service name for Internet Explorer profile

The authentication service used by the Internet Explorer profile is derived from the webpage URL to which it is connected. The Connect To dialog contains the IP address/host name of the page that is connected to in the dialog window title. The profile uses that part of the URL only for the authentication service name. For the example shown previously, the authentication service name is `10.150.22.2`.

Web Single Sign-On using Firefox browser

Figure 5-97 shows the Basic Authentication prompt dialog for Firefox.

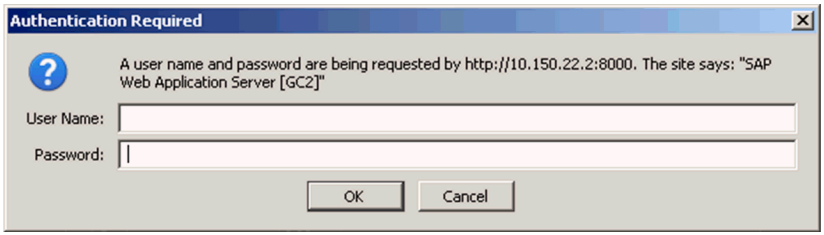

Figure 5-97 SAP WebGUI Basic Authentication with Mozilla Firefox

The IBM Security Access Manager for Enterprise Single Sign-On profile for Mozilla Firefox Basic Authentication is called `sso_site_wnd_firefox`.

The User Name and Password fields of this dialog are not able to be specified by using a Windows XPath, so the Firefox profile uses a *Show a dialog to capture logon credentials* action to prompt for and capture logon credentials. The Firefox profile has these characteristics:

- Injecting SAP logon credentials by using the Firefox profile

  From the state_start state, the profile transitions to *Basic Auth Shown* via a *Window is activated* trigger, which triggers when the Firefox Basic Authentication prompt dialog is displayed. During this state transition, stored credentials are injected into the Basic Authentication prompt dialog. There is only one instance of this trigger in the Firefox profile to handle the Basic Authentication prompt dialog that is presented in English.

  The following line is the Windows XPath signature for this trigger:

  ```
  /child::wnd[@title="Authentication Required" and
  @class_name="MozillaDialogClass"]
  ```

Click **OK** to submit the injected credentials to the SAP application for authentication. There is no validation of the correct logon with the injected credentials.

- Capturing SAP logon credentials by using the Firefox profile

  From the state_start state, the profile transitions to *Basic Auth Shown* via a *Window is activated* trigger, which triggers when the Firefox Basic Authentication prompt dialog is displayed. Credential injection is attempted, but if credentials are found in the Wallet, the transition from *Basic Auth Shown* state to the *Basic Auth Inject User* state causes a *Show a dialog to capture logon credentials* action to prompt for and capture logon credentials.

  When **OK** on the "IBM Security Access Manager for Enterprise Single Sign-On AccessAgent" prompt dialog is clicked, the entered credentials are copied to the Basic Authentication prompt dialog and are stored in the IBM Security Access Manager for Enterprise Single Sign-On Wallet. There is no validation of the correct logon with the injected credentials.

  The authentication service used by the Firefox profile is derived from the webpage URL that is connected. The Authentication Required dialog contains the IP address/host name of the page connected to in the dialog window text. The profile uses that part of the URL and port component of the URL for the authentication service name. For the example shown previously, the authentication service name is `10.150.22.2:8000`.

The configuration for the SAP application is complete.

The VMware View Client setup is described in Chapter 8, "Roaming desktop implementation" on page 259

5.4 Managing the deployed environment

After the installation and configuration of the base components, we look at managing and administering a deployed IBM Security Access Manager for Enterprise Single Sign-On environment.

Managing and administering the system involves the following tasks:

- Managing policies
- Managing users
- Logging

In a typical deployment scenario, you want to configure your policy before the AccessAgent installation, which was covered in 5.2.8, "Deploying AccessAgent" on page 149.

Further operational management topics are examined in Chapter 9, “Implementing operational requirements” on page 271

5.4.1 Managing policies

IBM Security Access Manager for Enterprise Single Sign-On uses policies to control the behavior of its components. Administrators have full control over policies, and users assigned to the help desk role have less privileged control over policies. Refer to Table 5-2.

Table 5-2 Policies and their scopes

Policy type	Administrator permission	Help desk permission	Policy scope
System policies	Full read/write	Read only	System-wide
Machine policies	Full read/write	Read only	Machines
User policies	Full read/write	Full read/write	Users

Policies are created and modified to enforce the rules set by the business. Before production deployment, you must have all of your policies clearly defined as direct translations of the business security requirements. Modifying policy after deployment might be unavoidable, but the best effort must be made to define policies before the deployment to production.

System, machine, and user policies have unique and overlapping policy parameters. Certain policies, such as AccessAgent lock/unlock, desktop inactivity, and logon/logoff can be defined in more than one policy type. In a deployment where many policies are defined, it is possible that several of the granular policies overlap. In this case, use the `managepolicypriority.bat` command-line utility to manage policy priorities. For more information, see the “Setting policy priorities” section in the *IBM Security Access Manager for Enterprise Single Sign-On Administrator Guide Version 8.2*, SC23-9951-03.

5.4.2 Managing users

Administrators have full control over users, and help desk teams have limited control over users assigned to their particular help desk. The help desk role is delegated administration based on group membership. You can assign users to help desks individually or by assigning a user policy to one or more help desks. Any user that is a member of the user policy is automatically under the delegated control of the help desks assigned to that policy. When planning for policy design,

ensure that you clearly define which help desks have control over which groups of users.

Another aspect of user management is managing the revocation of users or their Wallets. When a user leaves the company, an administrator can revoke the user Wallet, the second authentication factor of the user, or the actual user. Revoking the user permanently disables the user account and prevents any user with the same name from being created. When a user is revoked, all of its audit data is retained in the database. Only when a user is deleted is all of its audit data also deleted from the database. Remember this rule when defining administrative policy, so that important audit data is not accidentally deleted.

5.4.3 Logging

Three types of logs are available to the IMS Server: *user*, *system*, and *administrator*. User and administrator logs track user, administrator, and help desk activity. The user and administrator logs are considered the audit logs, and are written to the IMS database. Administrators and help desk officers can access the audit logs for individual users. Only administrators can run full queries on audit logs, access the help desk logs, and generate reports on help desk and user activity.

The system logs are message and error logs for the IMS Server, primarily used for troubleshooting server issues and monitoring the system health. Remember that when troubleshooting IMS Server issues, always copy the system logs before restarting the IMS Server. *Restarting the IMS Server clears the system logs*.

5.5 Conclusion

IBM Security Access Manager for Enterprise Single Sign-On provides single sign-on to many client applications without a lengthy and complex implementation effort. IBM Security Access Manager for Enterprise Single Sign-On provides users with one password to log on to many applications on both the company network and the Internet.

AccessAdmin simplifies administration by recognizing and configuring applications for sign-on automatically with minimal effort by the administrator. Enterprise users gain single sign-on while connected to or disconnected from the corporate network.

6

Password self-services implementation

In this chapter, we describe more details about the technical implementation of the IBM Security Access Manager for Enterprise Single Sign-On environment.

First, we introduce the password self-service feature. After describing the technical prerequisites, we explain how to configure the password self-service. Finally, we describe how to manage and maintain the deployed solution.

6.1 Business requirements

If users at the cardio healthcare company forget their Microsoft Windows password, they must contact the IT support desk to have the support desk personnel reset their password on their behalf after performing the necessary security checks. IBM Security Access Manager for Enterprise Single Sign-On overcomes this problem by providing the password self-service function of the product. Users that have a connection to the IMS Server can reset their own passwords by themselves.

6.2 Password self-service architecture

The IBM Security Access Manager for Enterprise Single Sign-On password self-service enables users to reset their primary authentication (IBM Security Access Manager for Enterprise Single Sign-On password or desktop password) from any workstation based on a challenge-response process. All questions are customizable and configurable. When the IBM Security Access Manager for Enterprise Single Sign-On password self-service is configured (no additional components must be installed), there is no need to call technical support and no waiting for an administrator to reset the password. Instead, the users provide secondary secrets that they set up during the sign-up phase of AccessAgent.

No additional components must be installed to use the password self-service function. Therefore, the architecture of the solution appears nearly the same as the basic architecture in 5.1.2, “Deployment architecture” on page 109, that uses AccessAgent on the Windows workstations.

Let us now look into the configuration details for the cardio healthcare company deployment, in particular, two common workflows for password self-services. The steps described in this chapter build on the base installation of IBM Security Access Manager for Enterprise Single Sign-On that was explained in Chapter 5, “Base installation and configuration” on page 107.

6.3 Implementing password self-service

When using the IBM Security Access Manager for Enterprise Single Sign-On password self-service, different workflows can occur. The workflow differs depending on whether the IMS Server is available when executing the password self-service request. When the password self-service feature is disabled but the

user still wants to reset the password, another workflow is triggered. Before enabling the password self-service functionality, we must set up the self-service secret questions.

6.3.1 Setting up the self-service questions

When the user wants to use the password self-service function, a series of questions must be answered in preparation. The questions are predefined and managed by the administrator by using the AccessAdmin console.

A list of predefined questions are part of the standard installation of the IMS Server:

- What's your favorite color?
- What's your favorite fruit?
- What's your mother's maiden name?
- Who's your favorite author?
- Who's your favorite composer?
- Who's your favorite person from history?

Configuring challenge-response questions

Challenge-response questions are prepared by the administrator. When the cardio healthcare company determines the set of questions, they must configure them into IBM Security Access Manager for Enterprise Single Sign-On.

The cardio healthcare company wants to include the following question in the Sign-up policies:

```
What's your favorite sport?
```

Follow these steps to configure system questions in the AccessAdmin console:

1. Open a web browser and enter the following web address:

   ```
   https://esso.cardio.example.com/admin
   ```

   This address directs the user to the IMS Server start page, as shown in Figure 6-1 on page 202. This page prompts for user name and password to access AccessAdmin. Provide the administrative user name in the User name text box and password in Password text box, and then click **Logon** to log in to AccessAdmin.

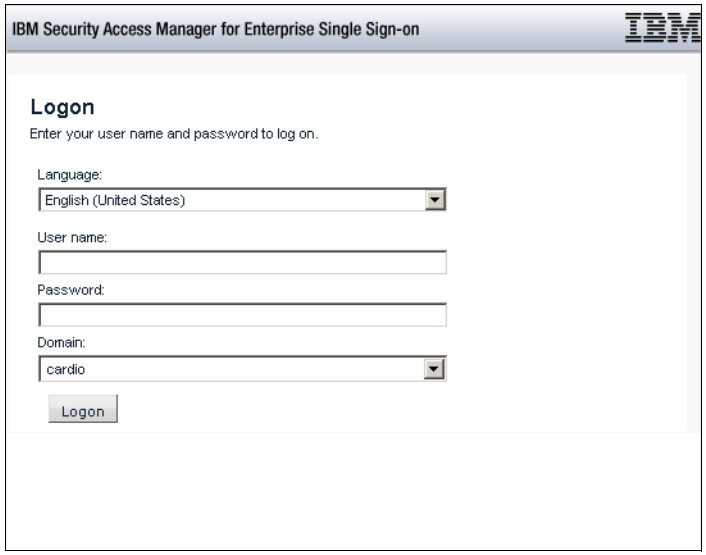

Figure 6-1 AccessAdmin Logon page

2. On the successful logon, you are presented with the AccessAdmin portal, as depicted in Figure 6-2 on page 203.

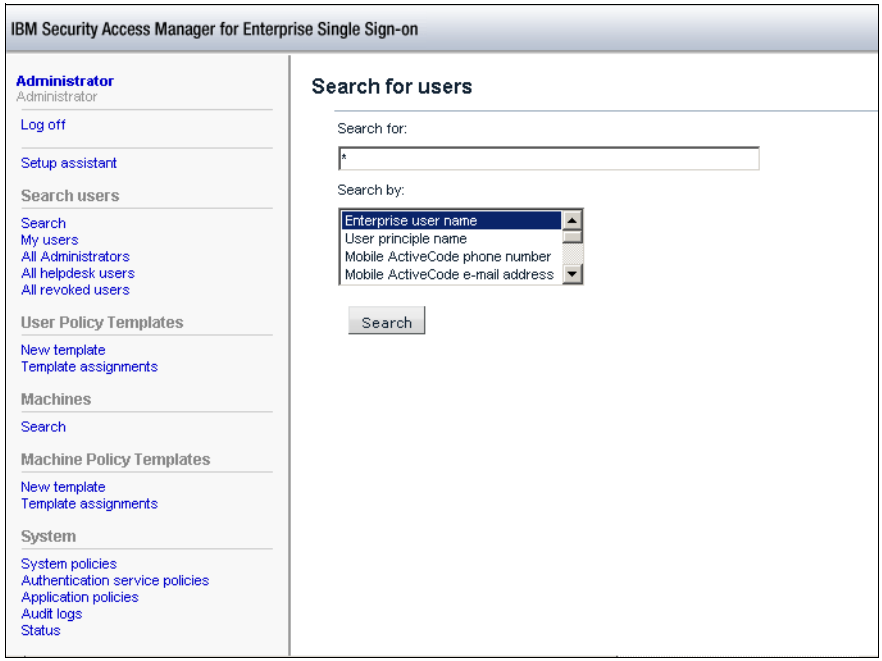

Figure 6-2 AccessAdmin portal page

3. Self-service-related configuration items are under System policies; therefore, click **System policies** (under System on the left side of the window). The System policies are listed, as shown in Figure 6-3 on page 204.

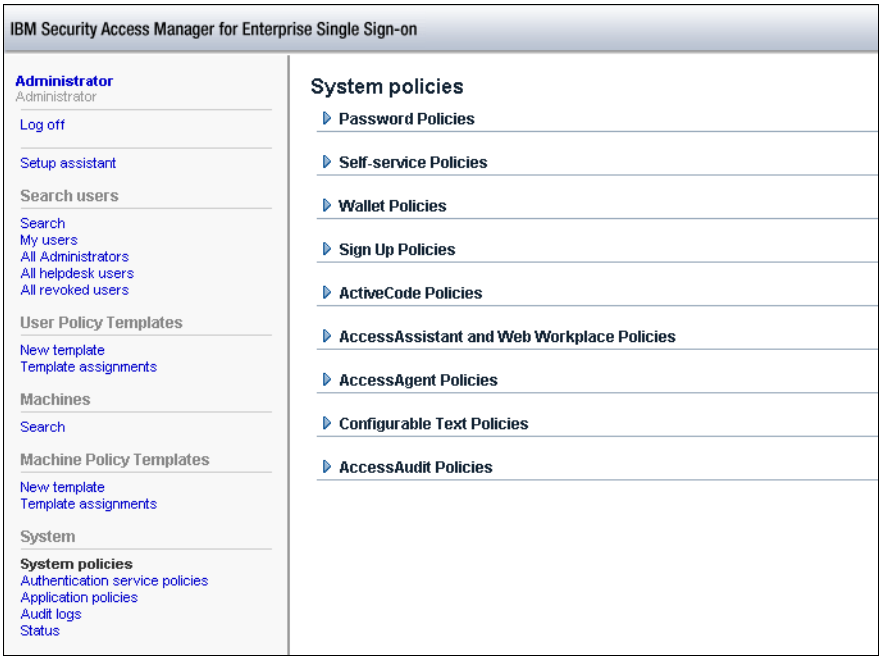

Figure 6-3 AccessAdmin System policies

4. Click the triangle beside **Sign Up Policies** to expand the list. The default secret questions are listed, as shown in Figure 6-4.

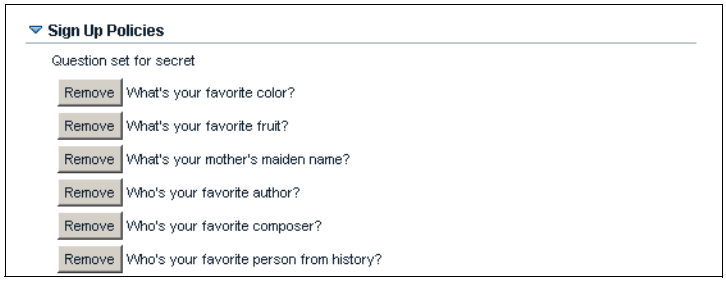

Figure 6-4 Default AccessAdmin self-service questions

5. In the text box, enter the following new question and then click **Add**, as shown in Figure 6-5 on page 205:

   `What's your favorite sport?`

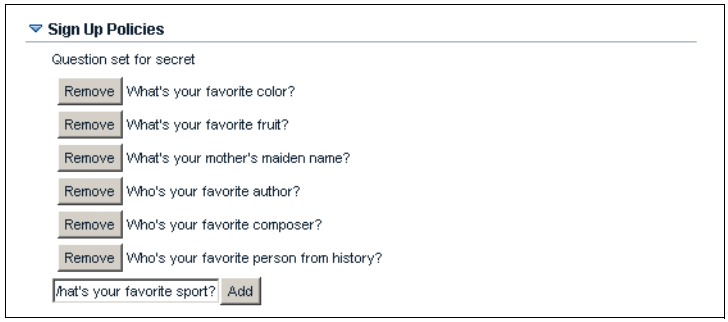

Figure 6-5 AccessAdmin self-service question configuration

The new question is added to the current Question set for secret, as shown in Figure 6-6.

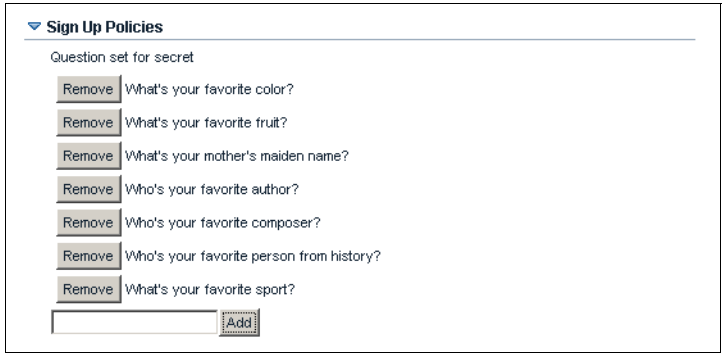

Figure 6-6 AccessAdmin self-service question configuration 2

6. If not already done, set “Prompt user to register additional secrets for self-service during sign-up?” to **Yes**, as shown in Figure 6-7 on page 206.

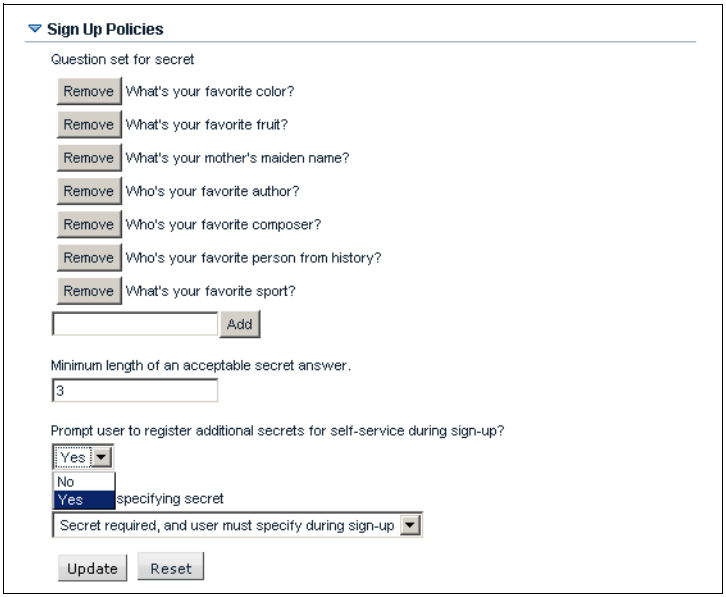

Figure 6-7 AccessAgent Sign-up Policy

7. If necessary, you can add more questions. After the self-service questions are configured, remember to click **Update** to apply any changes to the IMS Server.

After we configure our challenge-response questions, we are ready to enable the password self-service functionality.

6.3.2 Enabling the password self-service function

The password self-service can be disabled or enabled by system policy by using the AccessAdmin GUI. Depending on the status of the self-service feature, the password workflow differs.

Follow these steps to enable the password self-service function:

1. Open a browser window and enter the web address:

   `https://esso.cardio.example.com/admin`

   This address directs you to the IMS Server start page, as shown in Figure 6-8 on page 207. It prompts for user name and password to access AccessAdmin. Provide the administrative user name in the User name text box and the password in the Password text box, and then click **Logon** to log in to AccessAdmin.

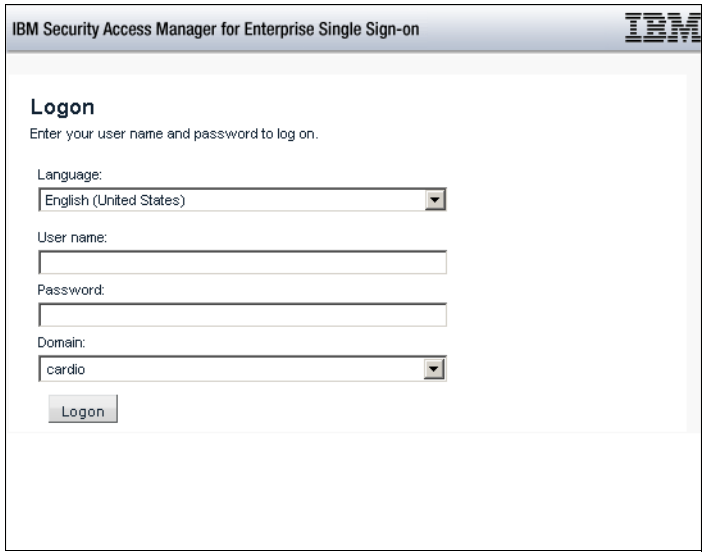

Figure 6-8 AccessAdmin start page

2. On the successful logon, the AccessAdmin portal opens (Figure 6-9).

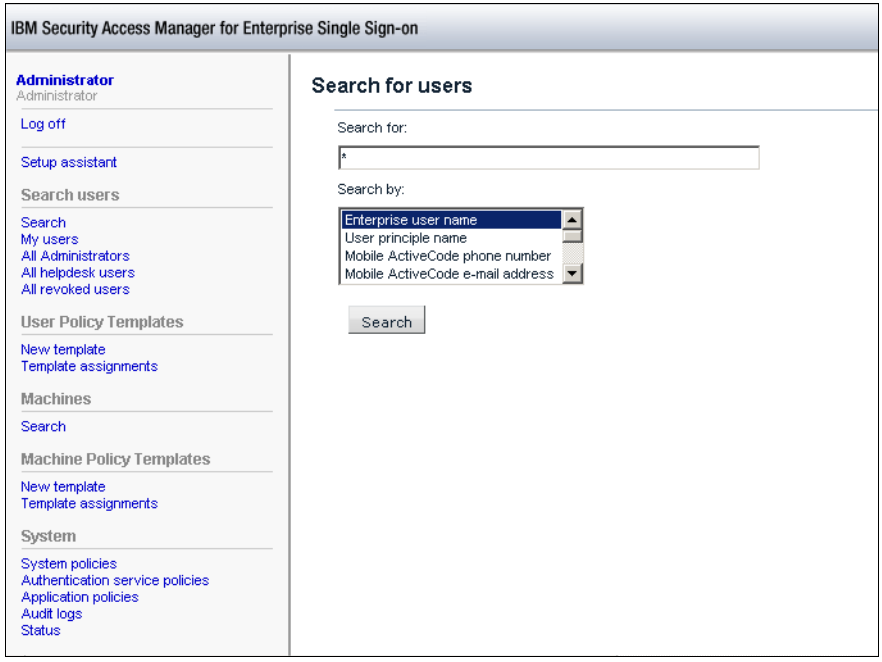

Figure 6-9 AccessAdmin portal

3. Self-service-related configuration items are under System policies; therefore, click **System policies** (under System on the lower left side of the window). The list of System policies is displayed; see Figure 6-10.

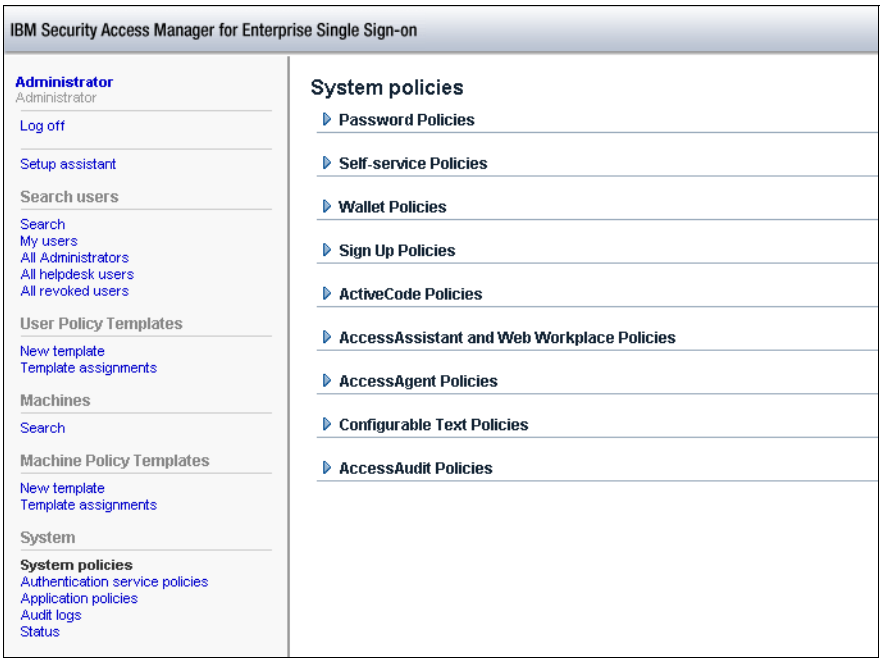

Figure 6-10 AccessAdmin system policies

4. Click **Self-service Policies** to expand the list. The panel expands and offers three choices, as shown in Figure 6-11 on page 209.

Figure 6-11 Self-service policies

5. Click **Self-service Password Reset Policies** to expand the section.
6. Ensure that “Enable self-service password reset?” is set to **Yes**. Ensure that “Maximum number of secret questions a user should register to enable self-service.” is set to 3. Ensure that “The number of secret questions a user needs to answer to use self-service” is set to 2, as shown in Figure 6-12. The sample values that ship with the default installation of the IMS Server are also shown in Figure 6-12.

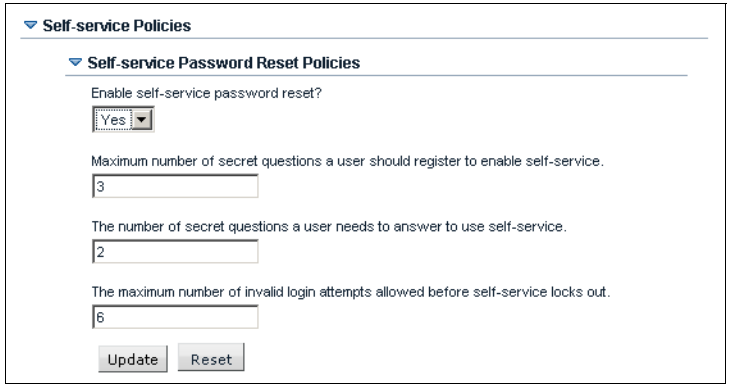

Figure 6-12 Self-service password reset policies definition

7. Now, the password self-service functionality is active. Click **Update** to apply the changes to the IMS Server.

6.3.3 User enrollment interview

Before the user can use the password self-service when necessary, the user must provide the correct answers to the questions that are asked. The questions in the enrollment interview are used to create the reset quiz that users take if they ever need to reset the IBM Security Access Manager for Enterprise Single Sign-On password. The answers provided in the interview are the answers that are used to verify the user's identity.

The easiest way to complete the enrollment interview is during the sign-up phase for a new user.

An example of the enrollment interview is demonstrated in the following steps:

1. On a Windows workstation with AccessAgent installed, go to the logon window, as shown in Figure 6-13.

Figure 6-13 AccessAgent logon dialog

2. Click **Sign Up** to start the enrollment of a new user, and enter the User name and Password of the cardio domain of the enterprise directory (Figure 6-14 on page 211). Then, click **Next**.

Figure 6-14 AccessAgent user enrollment

3. Enter the Password for the new IBM Security Access Manager for Enterprise Single Sign-On user (Figure 6-15).

Figure 6-15 IBM Security Access Manager for Enterprise Single Sign-On password

4. The list of predefined secret questions that the administrator previously defined is displayed, as shown in Figure 6-16 on page 212.

   The cardio healthcare company selected "`What's your favorite sport?`" as the first question. We provide our answer in the Answer text box, and then click **Next**.

Figure 6-16 AccessAgent first secret question

5. Answer the second question. In the company testing phase, we decide to answer "`What's your favorite color?`" (Figure 6-17).

Figure 6-17 AccessAgent second secret question

6. If you want to register more questions (to make it more secure), you can select "Register more questions". The cardio healthcare company wants to use two questions as a minimum; therefore, we do not select this option.
7. Finally, click **Next** to finish the enrollment interview.

   AccessAgent continues, and after a few seconds, the Windows desktop is displayed (Figure 6-18 on page 213). Look for the IBM Security Access Manager for Enterprise Single Sign-On icon in the lower-right corner.

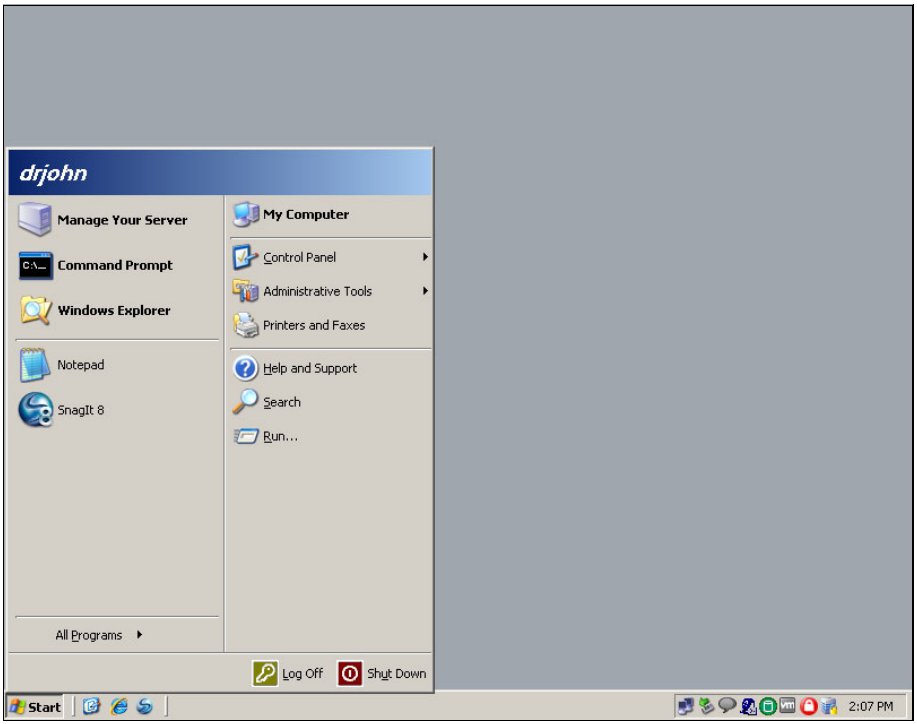

Figure 6-18 Windows desktop after Login

6.3.4 Executing a password reset

Several workflows can occur during a password self-service situation. If the user uses a user name and password for primary authentication, the following workflows for password reset occur:

- Online access to the IMS Server exists, and password self-service is enabled.
- No access exists to the IMS Server, or password self-service is disabled.

Let us take a closer look at both of these workflows.

Online access to the IMS Server exists and password self-service enabled

If AccessAgent can contact the IMS Server and the password self-service function is enabled, the user can process a password reset without contacting the help desk staff by providing the self-service credentials. Because the IMS Server can be contacted by AccessAgent, a password reset also updates the Wallet in the IMS Server.

To reset the password, follow these steps:

1. Go to the Windows workstation logon GUI and click **Reset Password**, as shown in Figure 6-19.

Figure 6-19 AccessAgent logon GUI

2. In the Reset Password dialog, as shown in Figure 6-20, enter the user name. During the testing phase, the cardio healthcare company uses drjohn, so we type `drjohn`. Then, click **Next**.

Figure 6-20 AccessAgent reset password

3. The user is then presented with a list of questions that were selected in 6.3.3, “User enrollment interview” on page 210. Select the first question and enter the answer, which, presumably, only the user knows (Figure 6-21 on page 215).

Figure 6-21 AccessAgent first secret answer

4. Click **Next** and answer the second question (Figure 6-22).

Figure 6-22 AccessAgent password reset second question

5. Click **Next**, and if everything works fine, that is, if the answers are correct, the password change dialog is presented, as shown in Figure 6-23 on page 216.

Figure 6-23 AccessAgent self-service password change

6. Click **Finish**. AccessAgent acknowledges the password reset, as shown in Figure 6-24.

Figure 6-24 AccessAgent successful password reset

No access to the IMS Server or password self-service disabled

If AccessAgent cannot contact the IMS Server to process a password self-service request, the user must contact the help desk to get an authorization code. There is no difference whether the password self-service is enabled or not. In offline mode, AccessAgent can access only local computer resources, in our case, the locally cached Identity Wallet of the user. If the IMS Server can be contacted by AccessAgent, as discussed in our scenario, a password reset updates both the Wallet on the IMS Server and the local Wallet.

If the connection to the IMS Server stops and there is no access, the involved parties, the user and the help desk, must work together to reset the user's password. In the following example, we show an offline password-reset workflow that involves both sides, the user activities and help desk activities.

User activities

The user activities consist of this workflow:

1. Go to the Windows workstation logon GUI and click **Reset Password**, as shown in Figure 6-25.

Figure 6-25 AccessAgent Logon GUI

2. Remember that a cached Wallet must exist. Click **Reset Password** and type the user name, as shown in Figure 6-26.

Figure 6-26 AccessAgent reset password

3. Click **Next**. If AccessAgent has no connection to the IMS Server, the message shown in Figure 6-27 is displayed and must be acknowledged; click **OK**.

Figure 6-27 AccessAgent password reset with no IMS Server connection

4. AccessAgent creates a request code that must be passed to the help desk to create a password reset authorization code. The user calls its help desk and tells them the request code, which in this case is `3RAH3HEX`, as shown in Figure 6-28.

Figure 6-28 AccessAgent password reset request code

5. After the user receives the authorization code from the help desk, the user enters it in the appropriate text field, as shown in Figure 6-29 on page 219.

Figure 6-29 AccessAgent password reset authorization code

6. Click **Next**. Answer the secret question. In our case, the system asks for our favorite sport, which is soccer. We type `soccer`. See Figure 6-30.

Figure 6-30 AccessAgent password self-service secret

7. Click **Next** and, if everything is correctly, you can specify a new password, as shown in Figure 6-31 on page 220. Because your AccessAgent has no connection to the IMS Server, this password is temporary.

Figure 6-31 AccessAgent password self-service new password

8. Click **Finish** to submit your new password. If your password is valid for a short time only (approximately one day to one month), AccessAgent states how long the temporary password is valid. See Figure 6-32.

Figure 6-32 AccessAgent temporary password

9. Click **OK**. AccessAgent immediately starts your desktop, as you can see in Figure 6-33 on page 221.

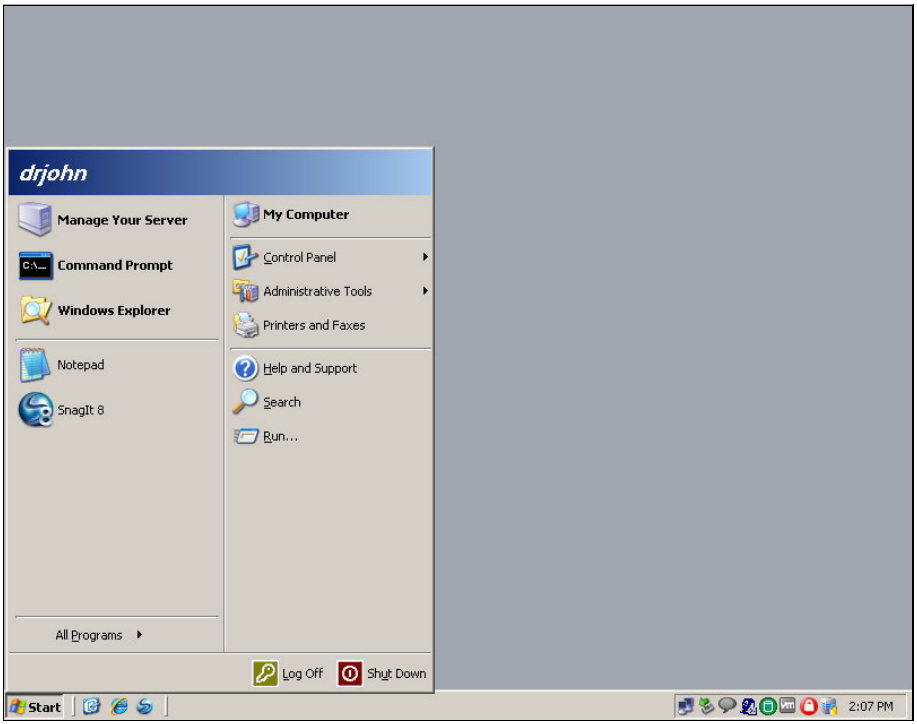

Figure 6-33 Windows desktop after password reset

Help desk activities

To support a user in resetting a Wallet password, the help desk staff must create an authorization code by using the AccessAdmin web GUI.

Follow these steps to create an authorization code:

1. Open a web browser and enter this URL:

   `https://esso.cardio.example.com/admin`

   If the IMS Server is installed, this address directs you to the IMS Server start page, as shown in Figure 6-34 on page 222. This page prompts you for the user name and password to access AccessAdmin. Provide the help desk *username* in the User name text box and the *password* in the Password text box, and then click **Logon** to log in to AccessAdmin. In our test case, `jeff` is the help desk user name.

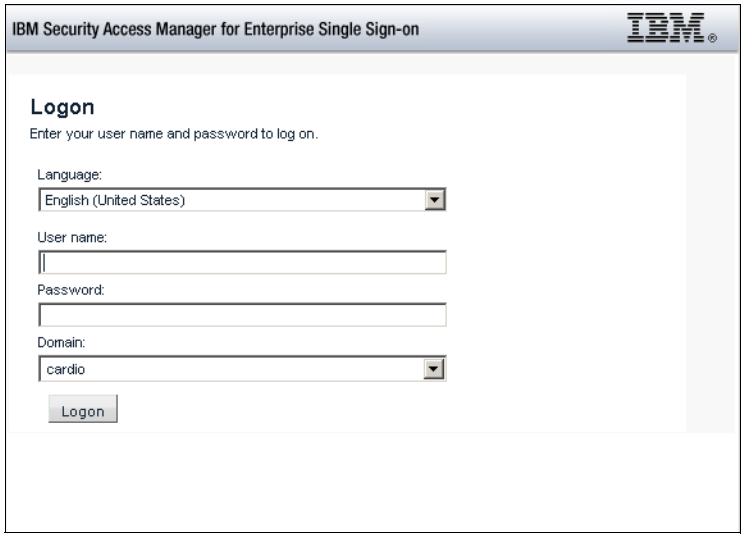

Figure 6-34 IMS Server web GUI

2. On the successful logon, you are presented with the AccessAdmin portal, as depicted in Figure 6-35.

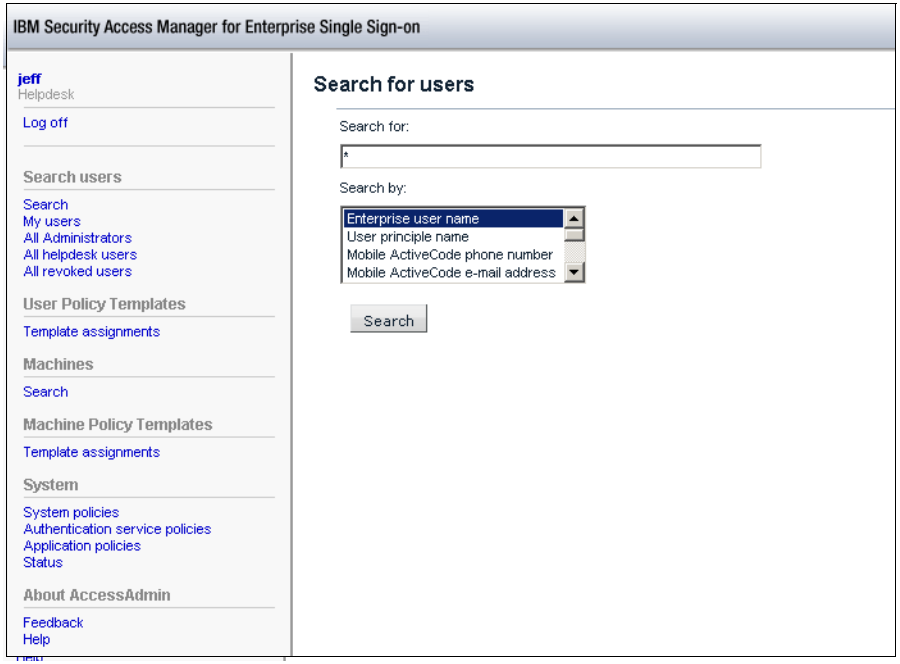

Figure 6-35 AccessAdmin portal page

3. To proceed with any user-related administration activities, enter the user's name `drjohn` in the text box, as shown in Figure 6-36.

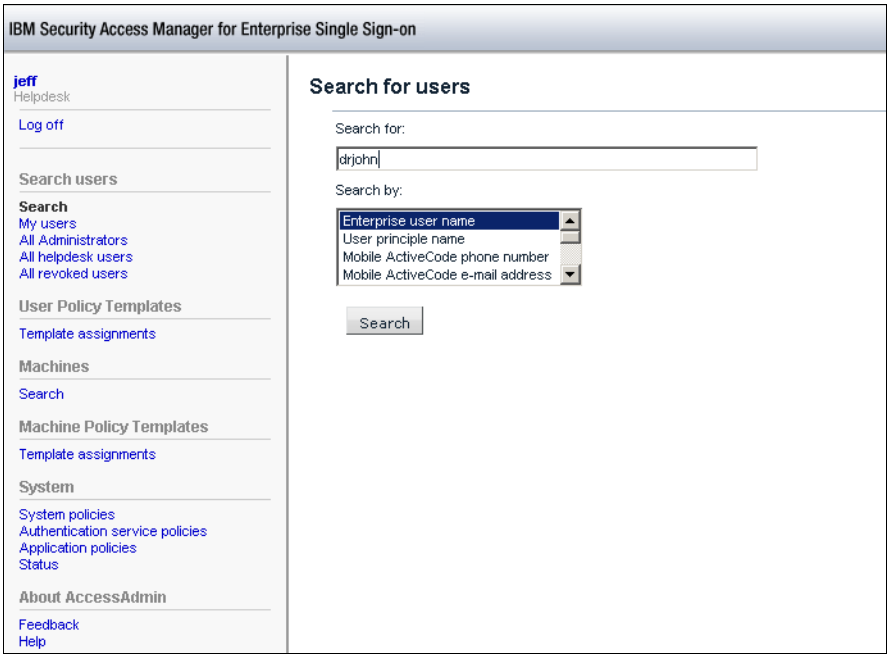

Figure 6-36 AccessAdmin User administration activities

4. Click **Search** to see the user information for itsouser2, which is shown in Figure 6-37 on page 224.

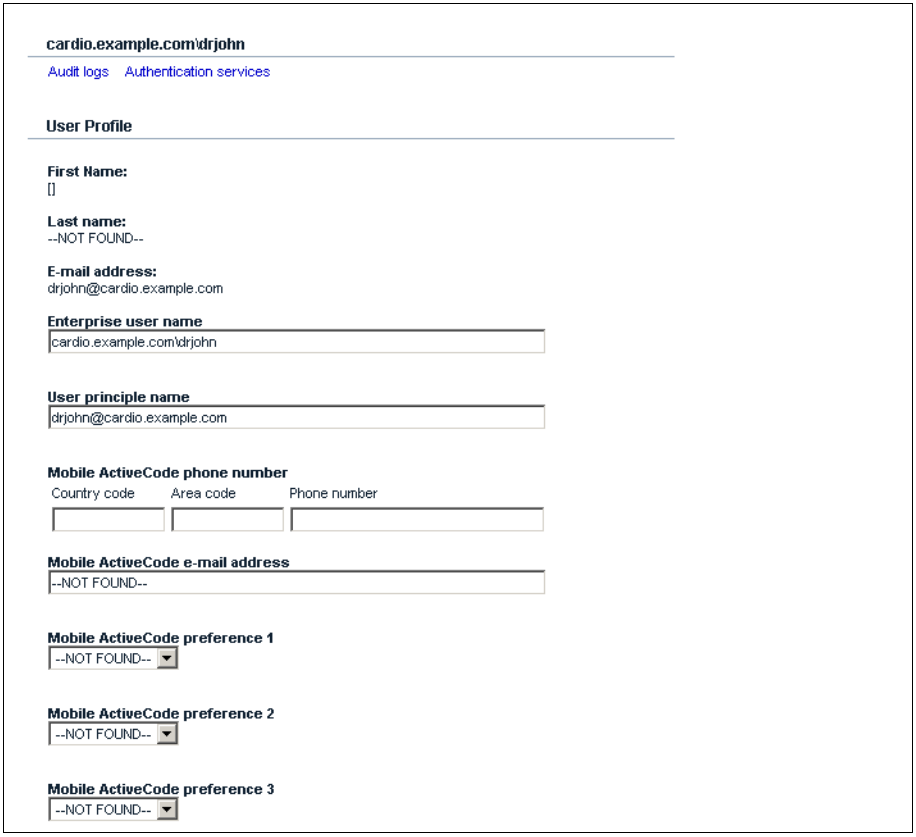

Figure 6-37 AccessAdmin user information

5. Click **Helpdesk Authorization** (Figure 6-38).

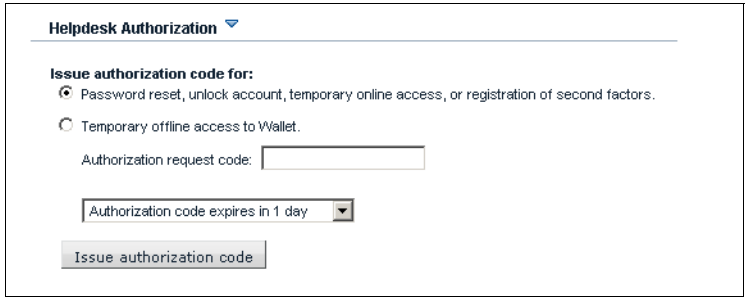

Figure 6-38 AccessAdmin Helpdesk Authorization

6. To create an authorization code for a password reset, select an expiration time (between one day and one month) and enter the request code that

appears on the AccessAgent GUI on the user's desktop. In the testing phase for the cardio healthcare company, the authorization request code is `3RAH3HEX`. Finally, click **Issue authorization code** (see Figure 6-39, and also Figure 6-28 on page 218).

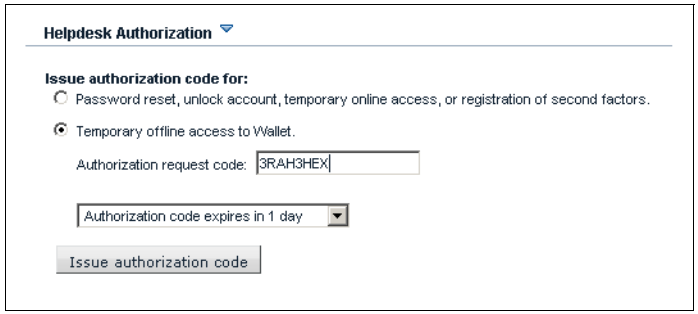

Figure 6-39 AccessAdmin authorization request

After several seconds, an authorization code is issued by the IMS Server (see Figure 6-40). This authorization code must be transferred to the user. The user needs the code to continue the password-reset self-service workflow.

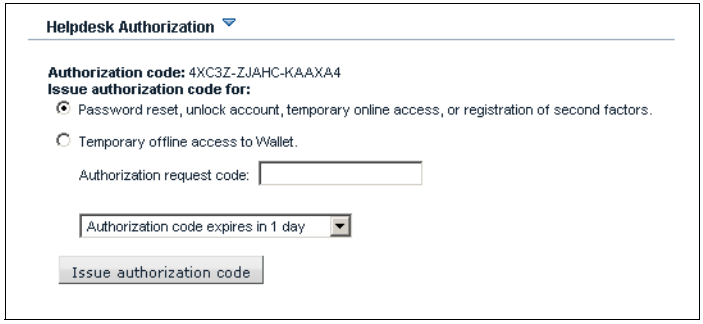

Figure 6-40 AccessAdmin-issued authorization code

6.4 Conclusion

The password self-service functionality enables users to reset their primary logon passwords without needing to call the help desk or find a special kiosk that is set up for that purpose, except when their workstation is not connected to the central IMS Server. The one-time enrollment is the only hurdle for the users before they can reap the benefits of this function, which is why you must educate users before deploying the solution within the enterprise.

Security administrators responsible for deploying the password self-service must take adequate time to formulate the questions before the initial deployment of the solution. Corporate security officers, human resources, and the CIO's office are several of the stakeholders that must review the questions.

When an organization, such as the cardio healthcare company, addressed these questions, the mechanics of deploying the solution discussed in this chapter allow for a smooth deployment of the function.

7

Strong authentication using RFID

The cardio healthcare company wants to offer a secure way of fast user switching to its medical staff. As we explained in 4.1.2, “Security and usability issues within the current infrastructure” on page 95, the medical staff, which uses the shared terminal for clients spread throughout the hospitals, needs a faster and more convenient way of logging on to the system. The medical staff often need to update a patient record with a few short comments before attending to the next patient, but they need to enter brief comments frequently each day.

The medical staff need to enter their user name and (complex) password numerous times per day to access their Virtual Desktop environment, which leads to frustration. The company committed to address this issue; however, it is not willing to compromise security. The cardio healthcare company opted to deploy radio frequency identification (RFID) badge readers to all shared terminal clients. This function enables the medical staff to link their RFID access badge to their single sign-on user name and password. The policy is designed to prompt the medical staff to present their RFID badge and their password one time each day. For the remainder of their shift, they can present their RFID badge to the reader and they are automatically logged on to their single sign-on Wallet.

Finger biometrics option: Another alternative that is available to the cardio healthcare company, and any organization that does not have a physical access system to use for strong user authentication, is *finger biometrics*. While it is designed to support enterprise-wide use, IBM Security Access Manager for Enterprise Single Sign-On finger biometrics-enabled authentication can be deployed to specific users and specific workstations (or other devices) only. In healthcare, for example, clinicians find this method of authentication convenient. They do not need to type a password repeatedly or handle a smart card or other second-factor token as they move between workstations.

Because a fingerprint cannot be “borrowed”, this approach also provides a higher level of certainty (and auditability for compliance purposes) that the person that accesses a patient record or prescribes a drug is correctly identified. IBM Security Access Manager for Enterprise Single Sign-On together with the BIO-key Biometric Service Provider finger biometric authentication option can support a wide range of fingerprint readers from all major manufacturers. This solution includes specialized readers for use with latex gloves.

For more information about this option, see “Configuring authentication to use fingerprint recognition” on page 425.

In this chapter, we describe the technical implementation of strong authentication with IBM Security Access Manager for Enterprise Single Sign-On.

7.1 Configuring machine and user policy templates

In this section, we describe how to configure the machine policy templates and the user policy template for the cardio healthcare company in a single configuration workflow.

7.1.1 Basic configuration by using the Setup assistant

First, we describe the basic configuration by using the Setup assistant. This configuration includes the selection of the workstation types that are used by the cardio healthcare company. Follow these steps:

1. Log on to the AccessAdmin interface in a web browser, as shown in Figure 7-1. The URL is `https://imsva/admin/faces/auth/login.xhtml` and `imsva` is the IMS Server, in our example.

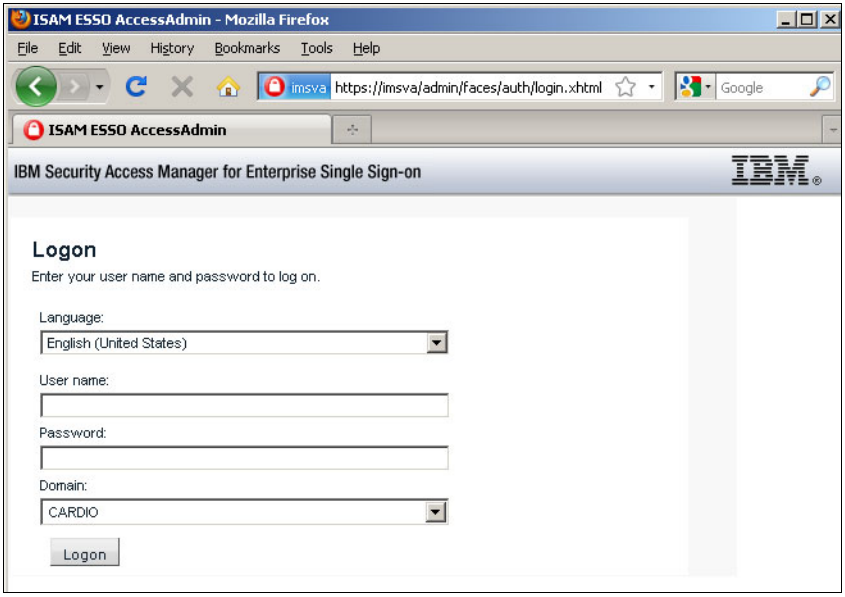

Figure 7-1 Log on to the AccessAdmin interface

2. Click **Setup assistant** and then click **Begin**, as shown in Figure 7-2 on page 230.

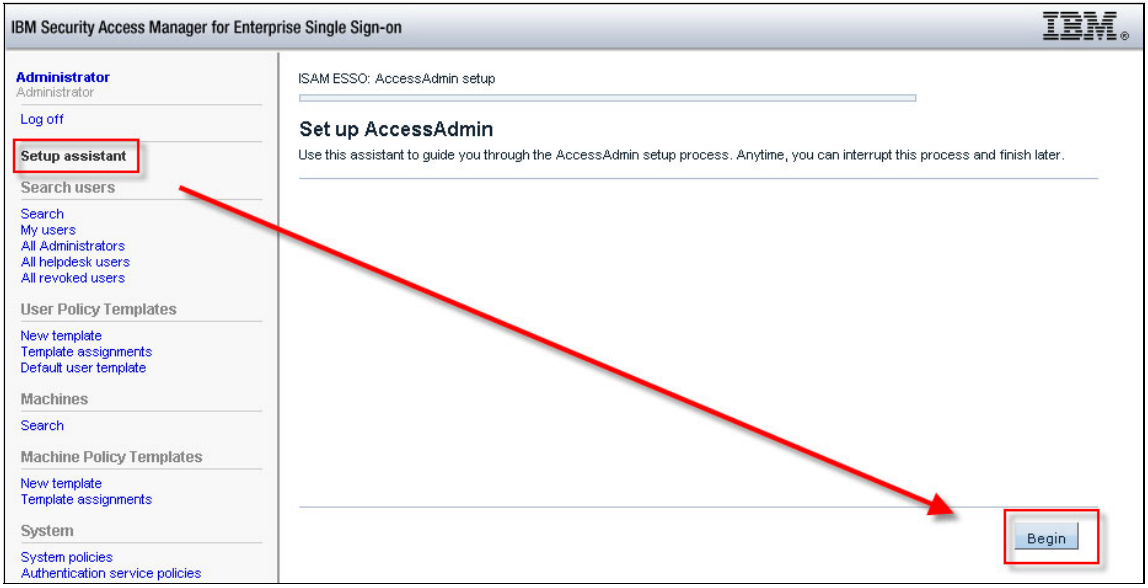

Figure 7-2 Start the Setup assistant

3. With Automatic sign-up, the users sign up automatically when they use AccessAgent for the first time. The self-service features offer functions, such as the self-service password reset. The cardio healthcare company decided to support both features for its users. Check **Enable automatic sign up** and click **Enable self-services features**, as shown in Figure 7-3 on page 231. Click **Next**.

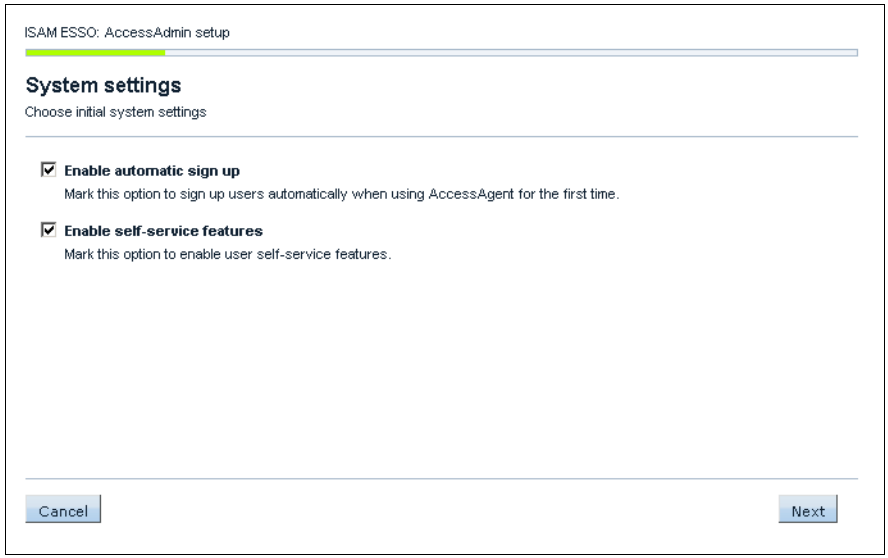

Figure 7-3 Enable system settings

4. Choose radio frequency identification cards (RFID cards) as the second-factor authentication for IBM Security Access Manager for Enterprise Single Sign-On, in addition to the Windows logon, as shown in Figure 7-4. Check **RFID card**, and click **Next**.

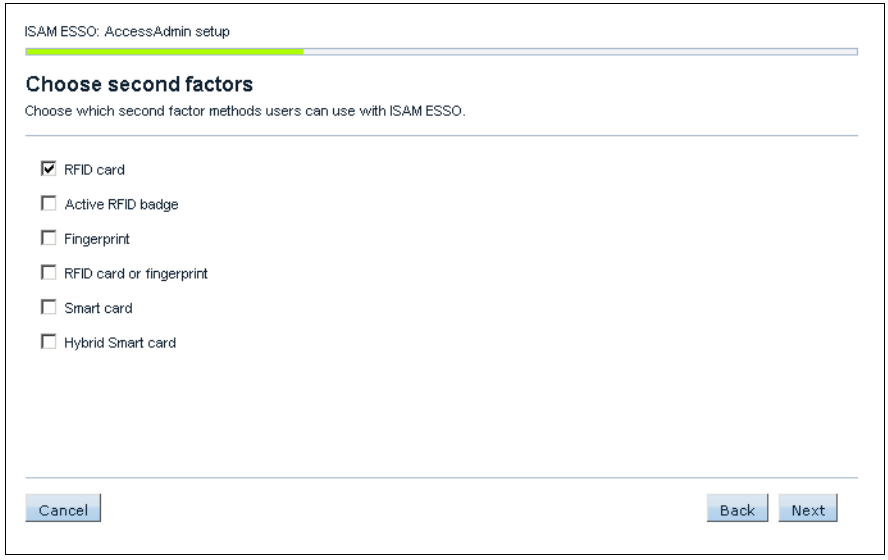

Figure 7-4 Selection of second-factor method

5. The cardio healthcare organization decided to implement personal workstations for healthcare personnel in the back office and shared workstations in the medical care units. We can configure both workstation sharing options in the same configuration process. Check **Support shared workstations**, click **Support personal workstations**, and click **Next**, as shown in Figure 7-5.

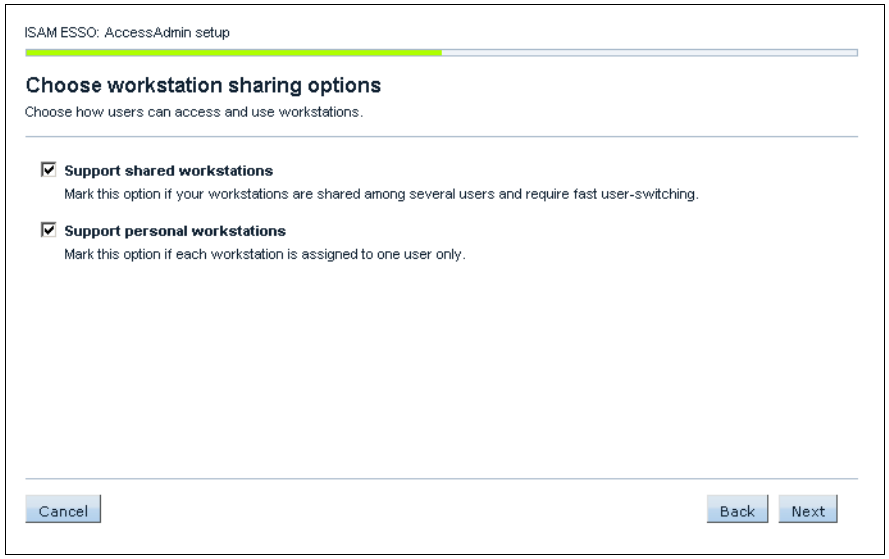

Figure 7-5 Select workstation sharing options

6. In Figure 7-6 on page 233, check **Use a shared desktop** only, and click **Next**. This function allows fast user switching by using the same underlying Windows account for different users; this concept is called a *shared desktop*.

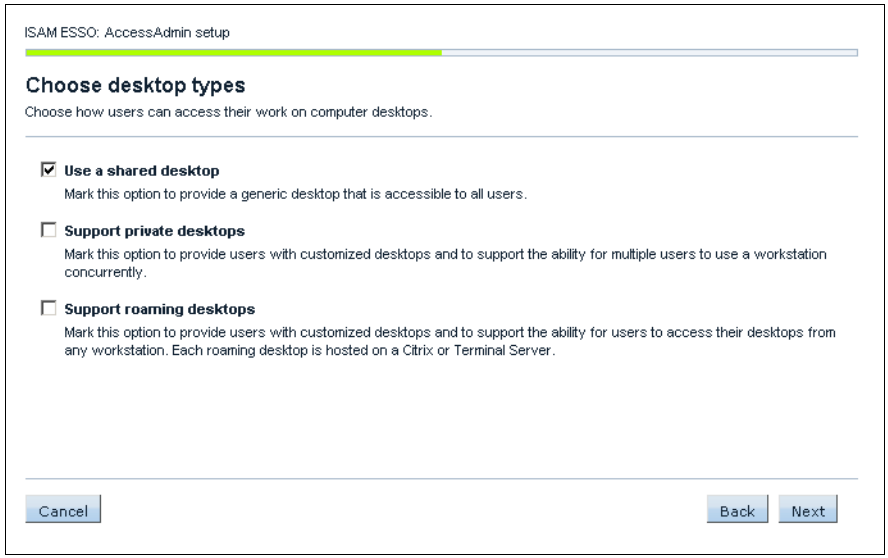

Figure 7-6 Choose desktop types

Workstation types:

- Personal workstation: The workstation is not shared with any other users.
- Shared workstation: A workstation is shared among users:
  - Shared desktop: A desktop scheme where multiple users share a generic Windows desktop.
  - Private desktop: Under this desktop scheme, users have their own Windows desktops in a workstation.
  - Roaming desktop: Under this desktop scheme, a user can disconnect from a desktop or application session at one client, log on to another client, and continue a desktop or application session at that new client.

7. In the next window (Figure 7-7 on page 234), do not select Enable AccessAgent for Citrix or Terminal Server. Click **Next**.

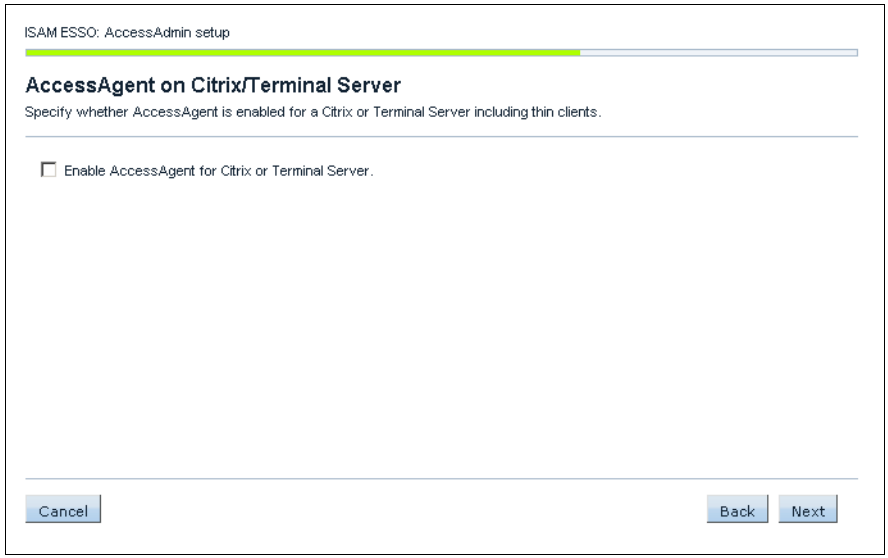

Figure 7-7 Enable Citrix or Terminal Server support

8. Now, you can choose a name for the user template, as shown in Figure 7-8 on page 235. The cardio healthcare company decided to name the user template “shared desktop user template” to emphasize the configuration for shared desktops. For Template name, type `shared desktop user template`. Click **Next**.

   There is only one user template needed for the cardio healthcare company for both shared and personal desktops currently. We show you later in this chapter how to configure the user template and the machine policy templates.

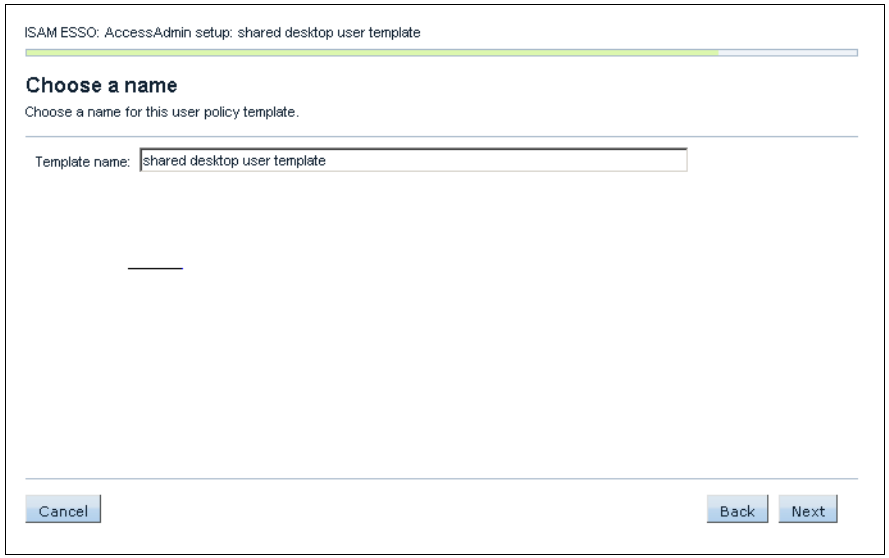

Figure 7-8 Choose a name for the user policy template

9. Check both **Password** and **RFID and password** to use for the Windows logon and click **Next** to enable RFID cards as the authentication factor for logon as shown in Figure 7-9.

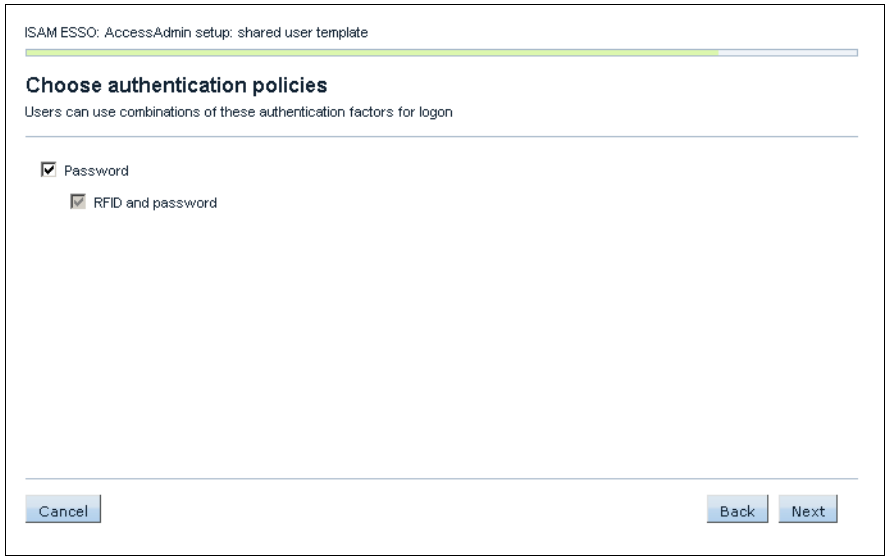

Figure 7-9 Choose support authentication factors for logon

10. The cardio healthcare company defined a policy that users can log on with their RFID card. They must log on with their password on a workstation initially but then can rely on their RFID card to log on during the remainder of the day. Type 480 to define 480 minutes (8 hours) and click **Next**, as shown in Figure 7-10.

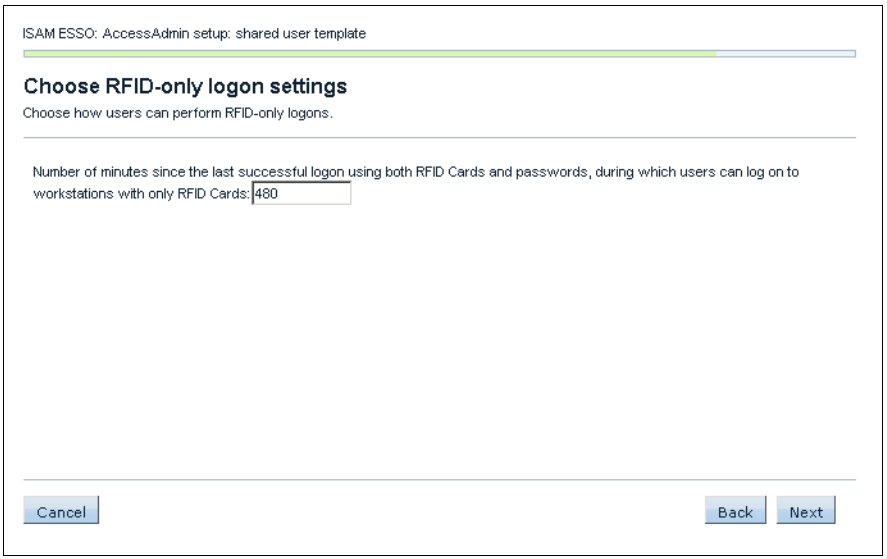

Figure 7-10 Define the time period for the RFID only logon

7.1.2 Configuring the personal workstation and RFID

Now, we look at the configuration details of the personal workstation machine policy template. We are still in the same configuration workflow.

In the following window, we see the two machine policy templates for personal and shared workstations that we defined. Before we click **Next** to finish the configuration process, we configure the details for the two machine policy templates.

First, we configure the machine policy template for personal workstations:

1. Click **Configure** next to the first line to configure Personal workstation, RFID, as shown in Figure 7-11 on page 237.

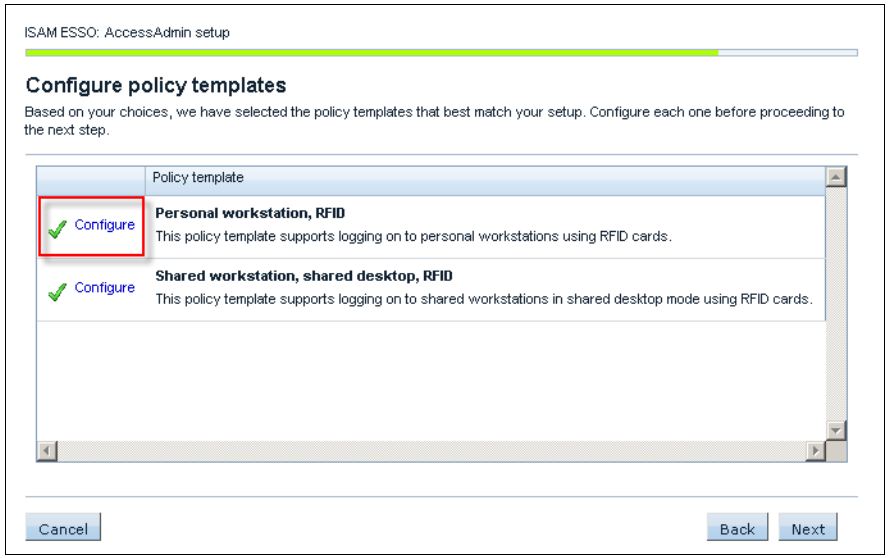

Figure 7-11 Configure policy templates

2. We can change the machine policy template name, as shown in Figure 7-12. Keep the name Personal workstation, RFID, and click **Next**.

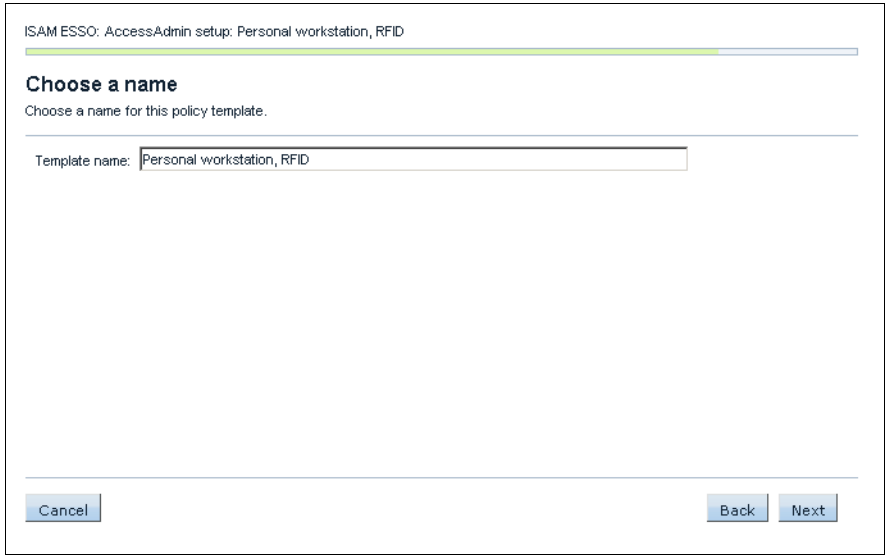

Figure 7-12 Choose policy template name

3. Check **RFID card only** to be the second-factor method for the personal workstation machine policy template, and click **Next** (Figure 7-13).

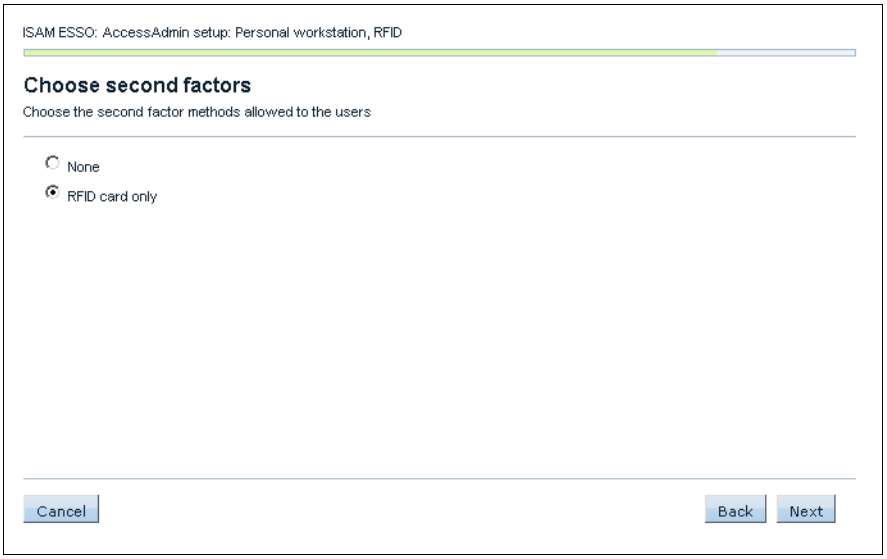

Figure 7-13 Allow RFID card only as second-factor method

4. IBM Security Access Manager for Enterprise Single Sign-On allows a transparent screen lock option. This option locks the screen and blocks user access, but it shows running applications as though the screen is not locked. The cardio healthcare company does not need this feature currently. Check **Normal screen lock**, and click **Next**, as shown in Figure 7-14 on page 239.

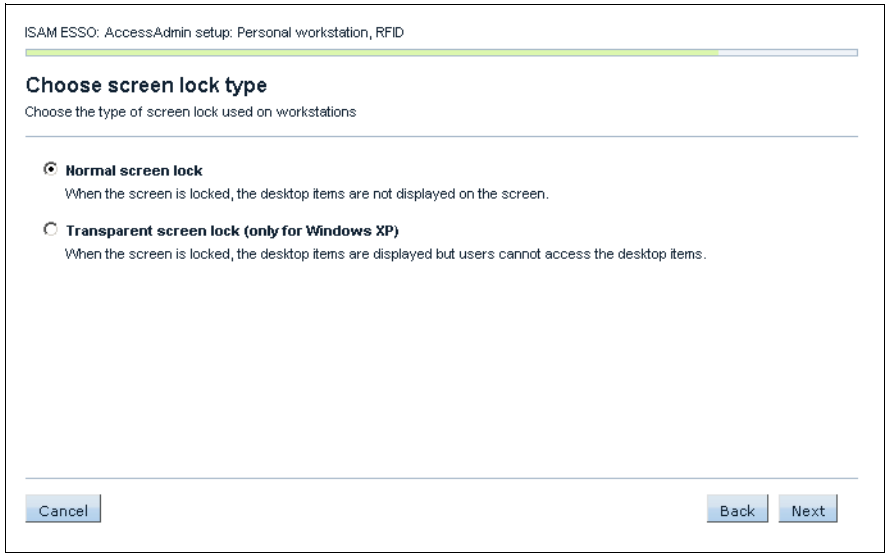

Figure 7-14 Choose a screen lock type

5. The cardio healthcare company decided to allow the RFID-only logon and RFID-only unlock to the personal workstations, therefore, the user does not need to provide a password. Check both **RFID-only logon** and **RFID-only unlock**, and click **Next** (Figure 7-15).

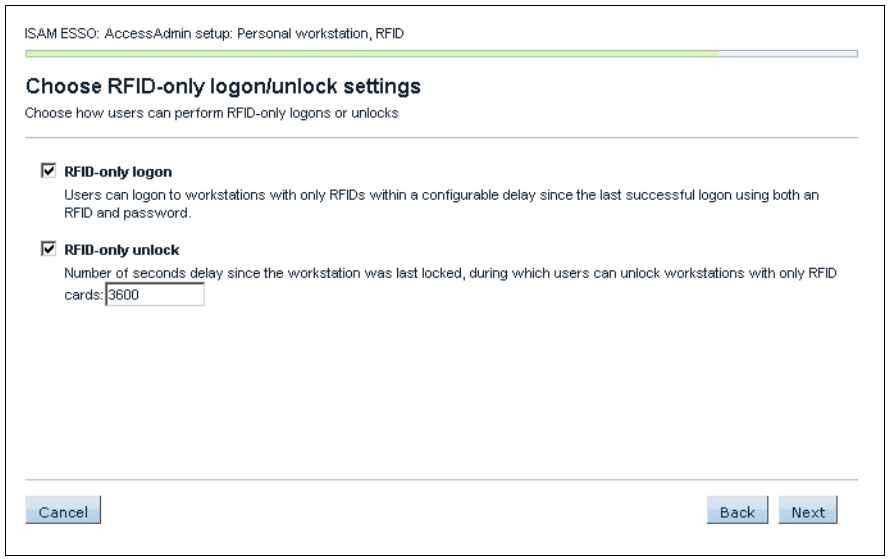

Figure 7-15 Choose RFID-only logon and RFID-only unlock settings

6. For personal workstations, the security policy of the company is defined to lock the screen after 30 minutes of inactivity. Type `30` in the input field, as shown in Figure 7-16, check **Lock computer**, and click **Next**.

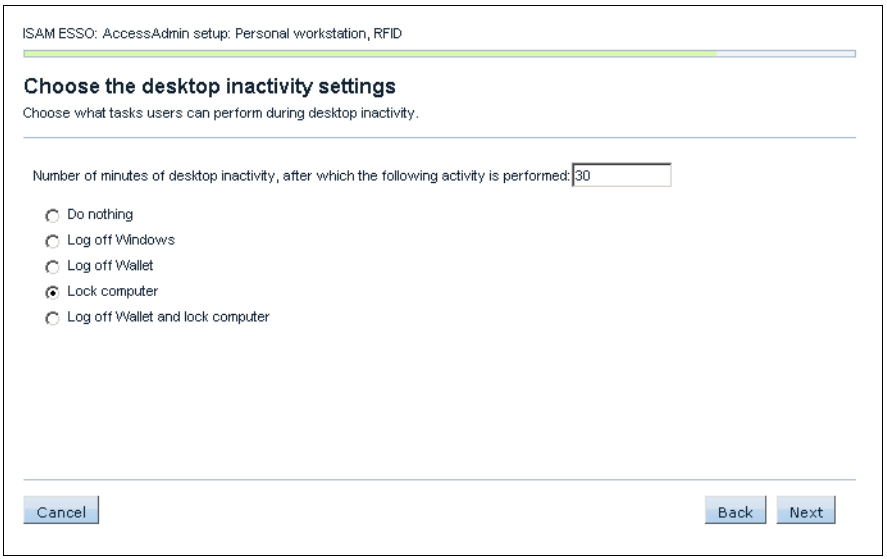

Figure 7-16 Choose the desktop inactivity settings

7. Select to use this machine policy template as the default for machines on which AccessAgent is newly installed. Check **Use this as the default template for machines** and click **Next**, as shown in Figure 7-17 on page 241.

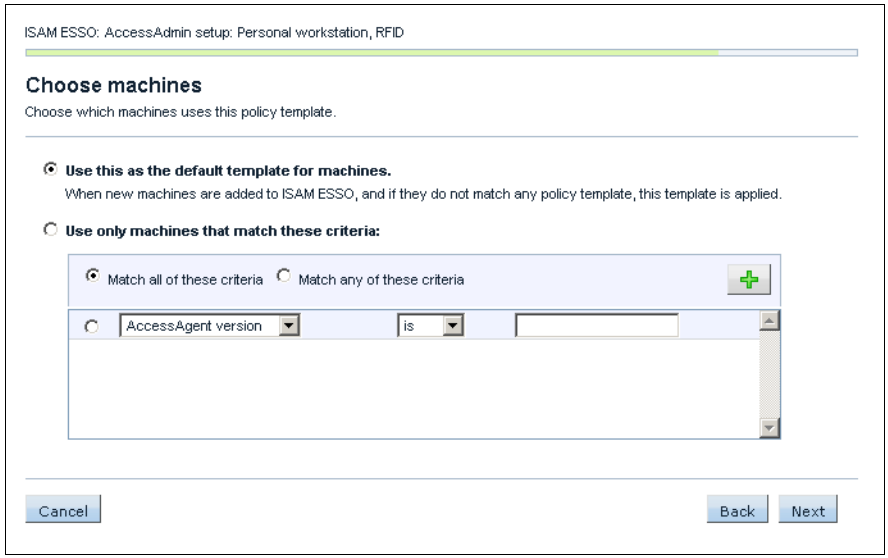

Figure 7-17 Define default machine policy template

Applying the policy: The default machine policy template applies automatically to machines on which AccessAgent is installed in the future. The default machine policy template does not apply to machines on which AccessAgent is installed before this configuration. You need to search for those machines in the AccessAdmin interface and apply the policy to them manually.

Restart machines: You might need to restart machines after applying specific machine policies to them.

7.1.3 Configuring shared workstations, shared desktops, and RFID

After we finish the configuration of the machine policy template for personal workstations, the overview is displayed, as shown in Figure 7-18 on page 242. We configure the machine policy template for shared workstations:

1. Click **Configure** next to the second line, Shared workstation, shared desktop RFID, and click **Next**.

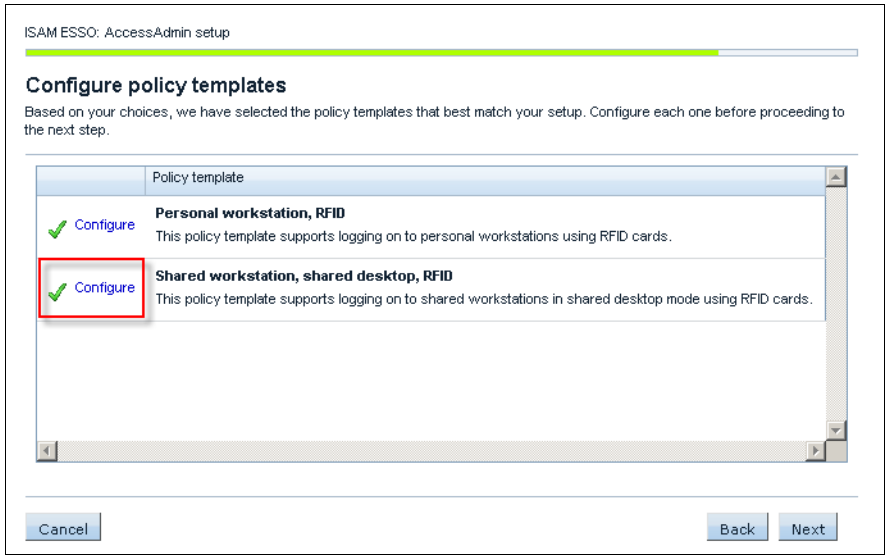

Figure 7-18 Configure machine policy templates

2. Define a template name `Shared workstation, shared desktop, RFID`, and click **Next** (Figure 7-19).

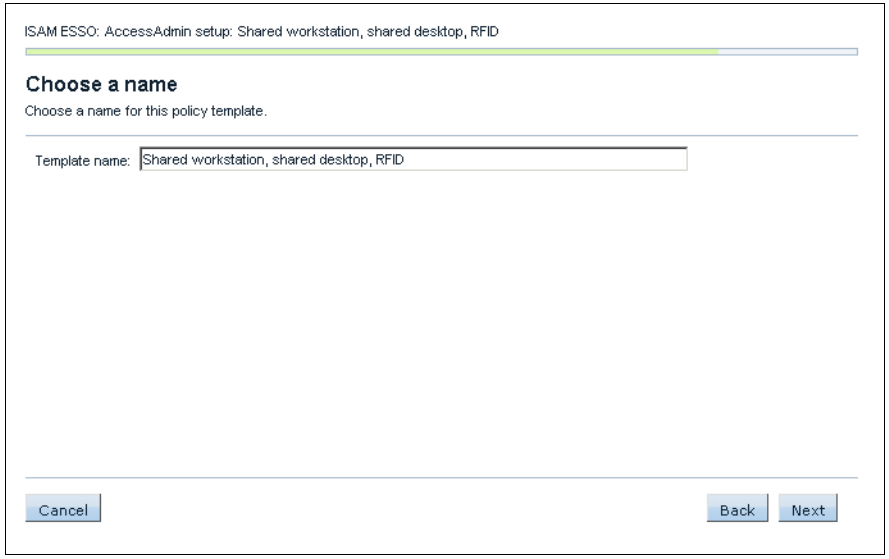

Figure 7-19 Define machine policy template name for shared workstations

3. Check **RFID card only** as the second-factor method, and click **Next** (Figure 7-20).

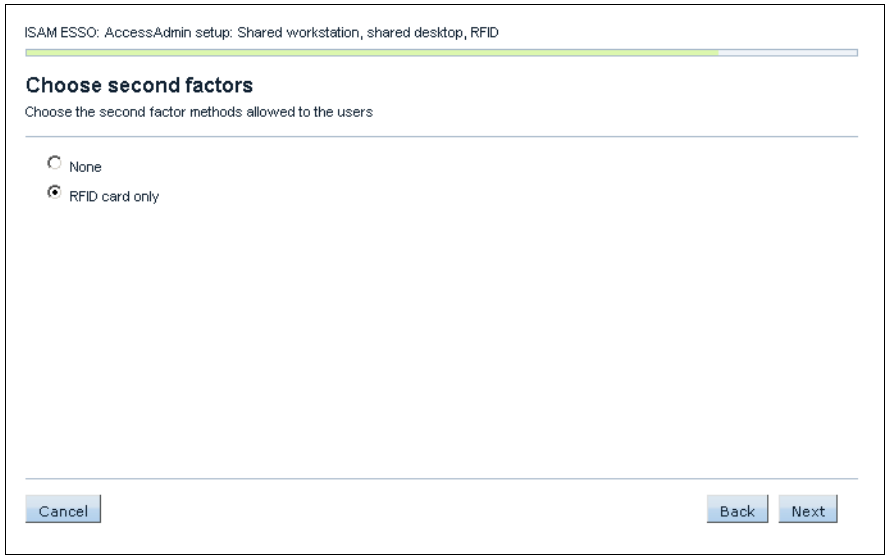

Figure 7-20 Choose RFID as the second-factor method

4. Keep the default screen lock type, **Normal screen lock**, and click **Next**, as shown in Figure 7-21.

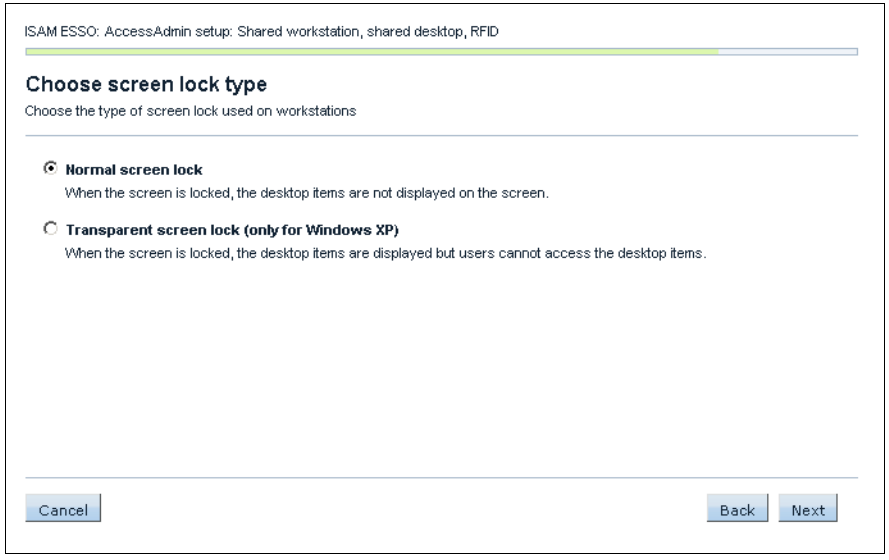

Figure 7-21 Choose screen lock type

5. For faster accessibility, we allow **RFID-only logon** and **RFID-only unlock** for shared workstations, too. Check both options, as shown in Figure 7-22, and click **Next**.

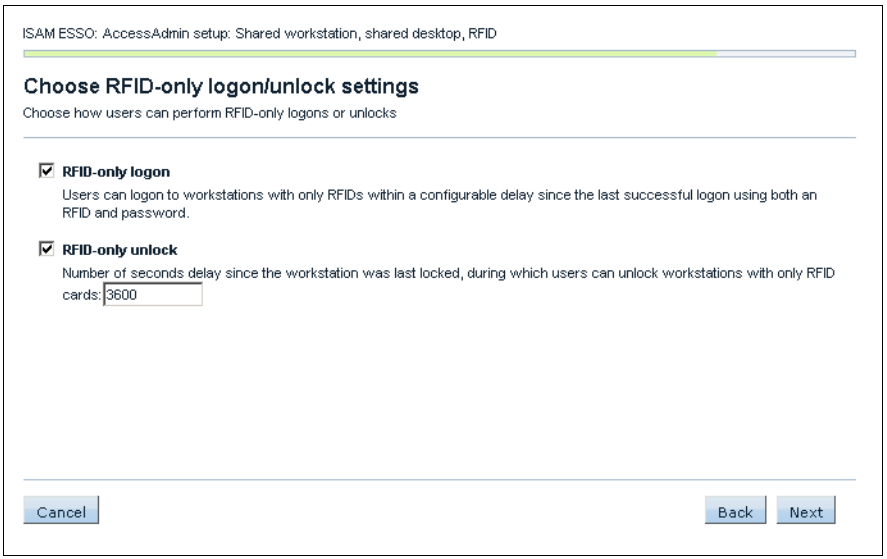

Figure 7-22 Choose to allow RFID-only logon and unlock

6. For personal workstations, we defined a desktop inactivity time of 30 minutes. For shared workstations, which are in semi-public areas, the cardio healthcare company decided to implement an inactivity time of only 5 minutes. Choose to **Lock computer** after **5** minutes of inactivity, as shown in Figure 7-23 on page 245, and click **Next**.

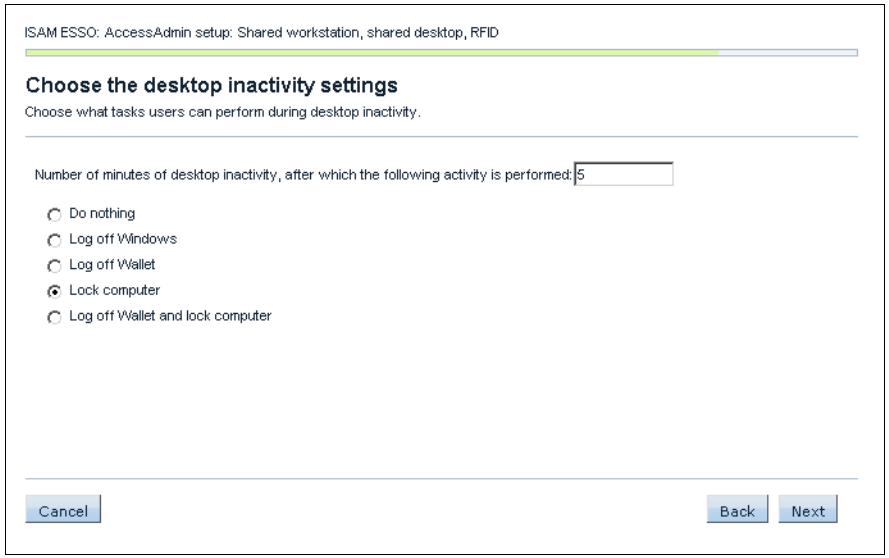

Figure 7-23 Choose desktop inactivity time of 5 minutes for shared workstations

7. Define a matching criteria to apply the policy template to machines, which are mostly Windows terminal clients, that enable access to the virtual desktop environment. The cardio healthcare company defined a naming convention: The host names of all terminal clients start with WT (as in Windows Terminal) as the host name. Follow these steps:

   a. Check **Use only machines that match these criteria**, as shown in Figure 7-24 on page 246.

   b. Select **Host name** from the pull-down menu.

   c. Select **is like** to match a substring.

   d. Type `WT` to match the Windows terminal clients whose host names start with WT.

   e. Click **Next**.

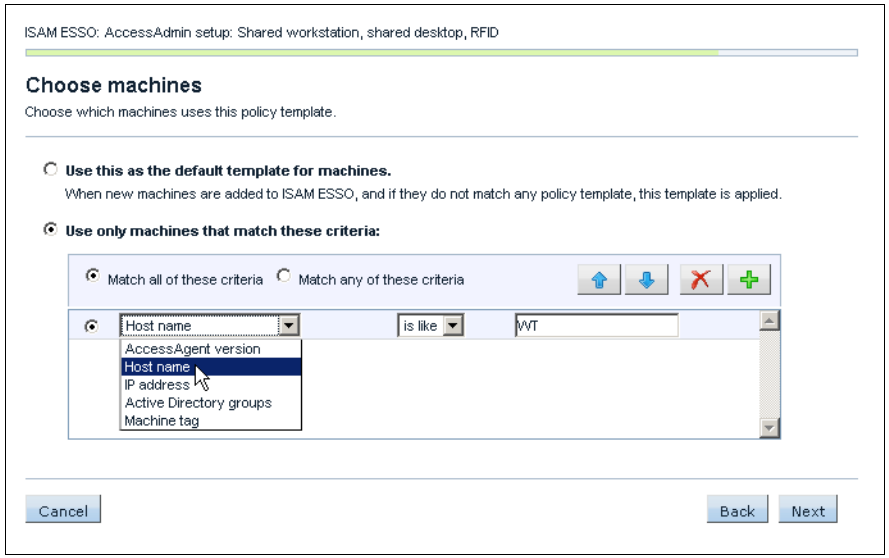

Figure 7-24 Define a matching rule for shared workstations

8. We are back to the Configure policy templates overview again. Now that the configurations for both machine policy templates are complete, click **Next** (Figure 7-25).

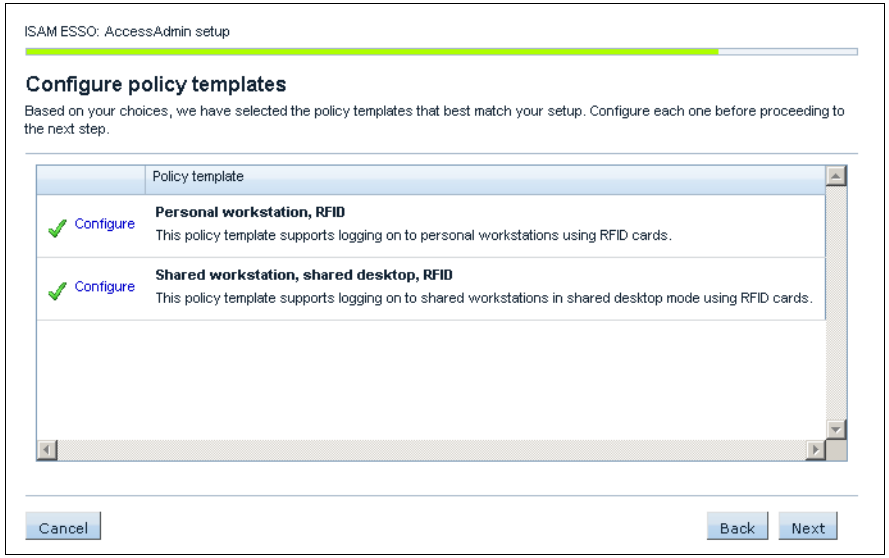

Figure 7-25 Policy template configuration overview

9. Click **Next** in Figure 7-26 to confirm the settings.

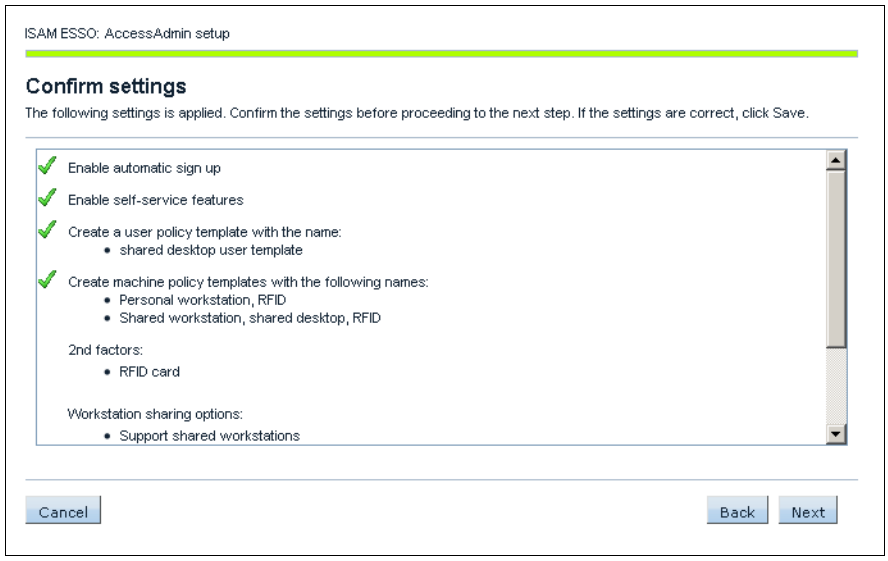

Figure 7-26 Confirm machine policy template settings

10. Click **Done** to finish the workflow, as shown in Figure 7-27.

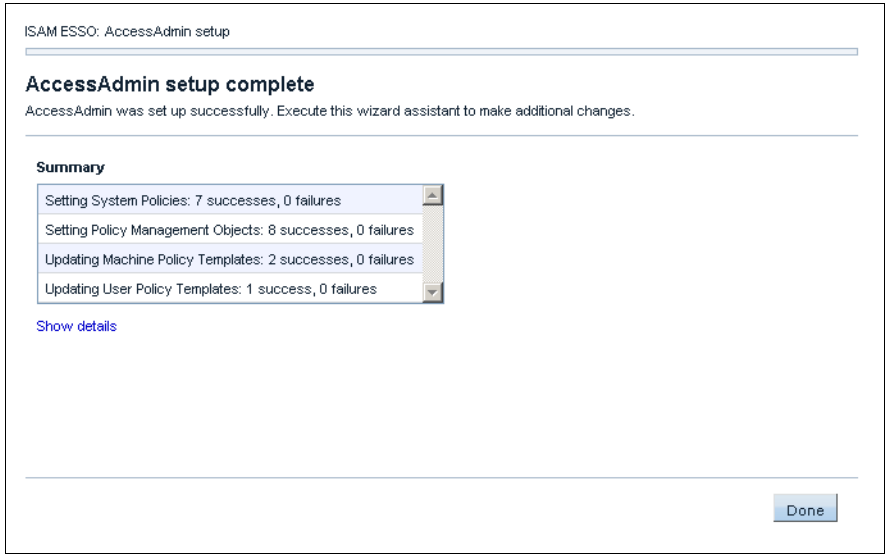

Figure 7-27 AccessAdmin setup complete

The basic configuration for the setup assistant is complete. The basic configurations of the machine policy template for personal desktops, the machine policy template for shared desktops, shared workstations, and the user policy template are complete.

7.1.4 Configuring details for the user policy template

Now, we configure the details for the shared desktop user template:

1. In the AccessAdmin GUI, click the newly created **shared desktop user template**, as shown in Figure 7-28 on page 249.
2. Expand **AccessAgent Policies**.
3. Expand **RFID policies**.
4. Change the value of the field “Confirmation countdown duration in seconds, for tapping same RFID on desktop” from 5 seconds to 0 seconds to allow immediate action.
5. Also, change the value of the field “Confirmation countdown duration in seconds, for tapping different RFID on desktop” from 5 seconds to 0 seconds to allow immediate action.

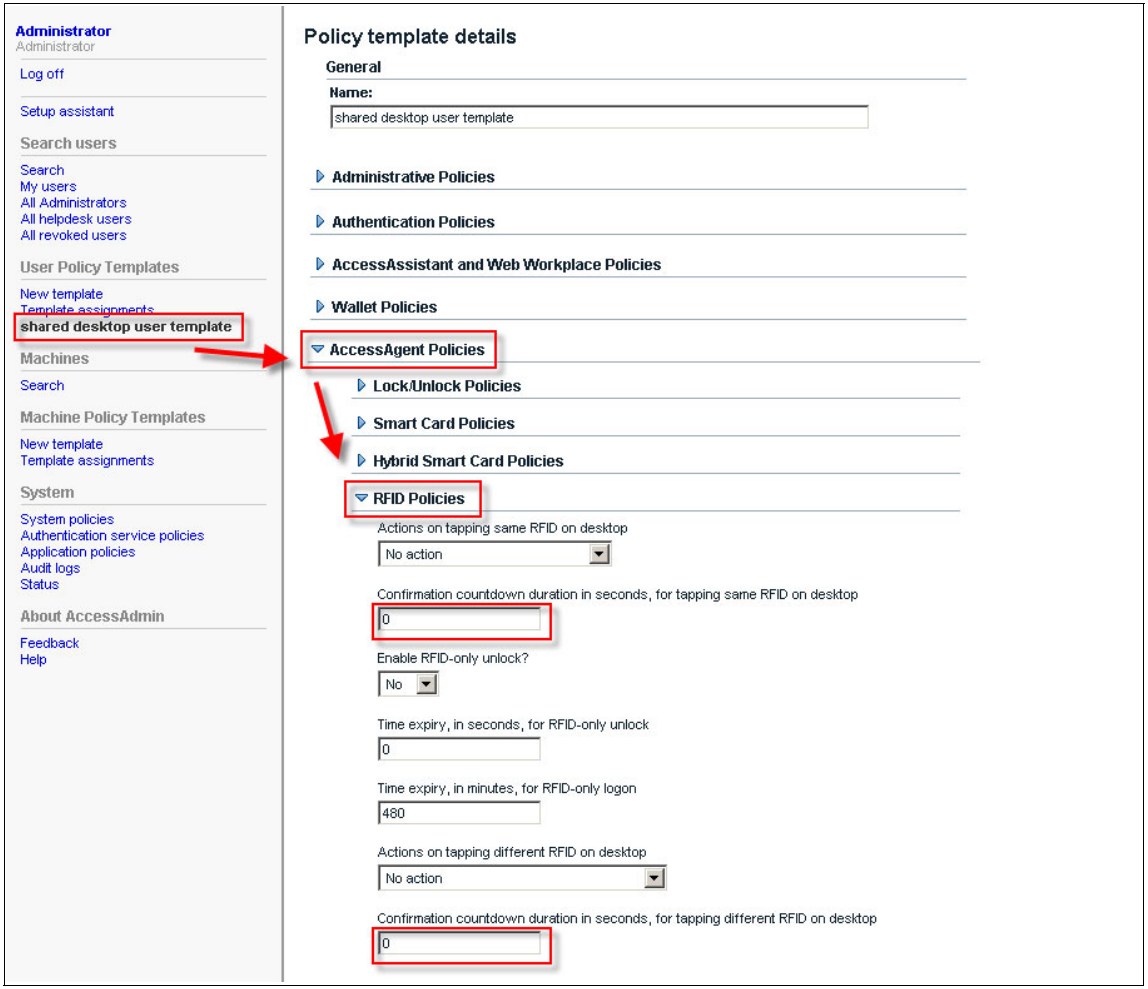

Figure 7-28 Start configuring user policy template and change RFID policy values

6. Scroll to the bottom of the page, and click **Update** to apply the policy changes, as shown in Figure 7-29 on page 250.

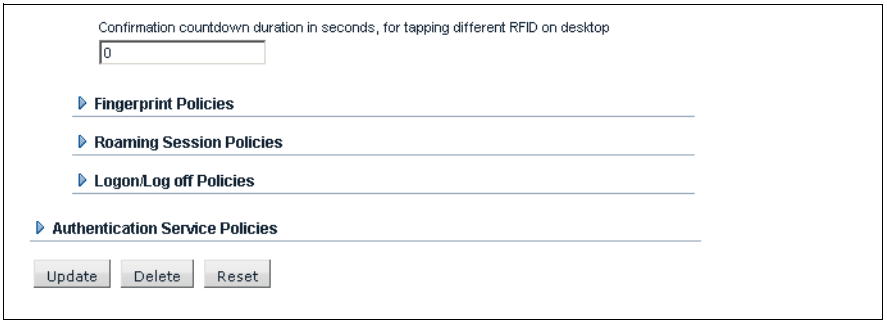

Figure 7-29 Apply policy changes

7. On the left panel, click **search** to search for a user, as shown in Figure 7-30.
8. The cardio healthcare company defined a search to use the user policy for all doctors. Search for `dr*`. Select **Enterprise user name**. Click **Search**, as shown in Figure 7-30.

Figure 7-30 Search for all users who are doctors

9. Click **Select all** to select all found users. Select the user policy template **shared desktop user template**. Click **Apply to selected results** to deploy the changed policy values to the selected users, as shown in Figure 7-31 on page 251.

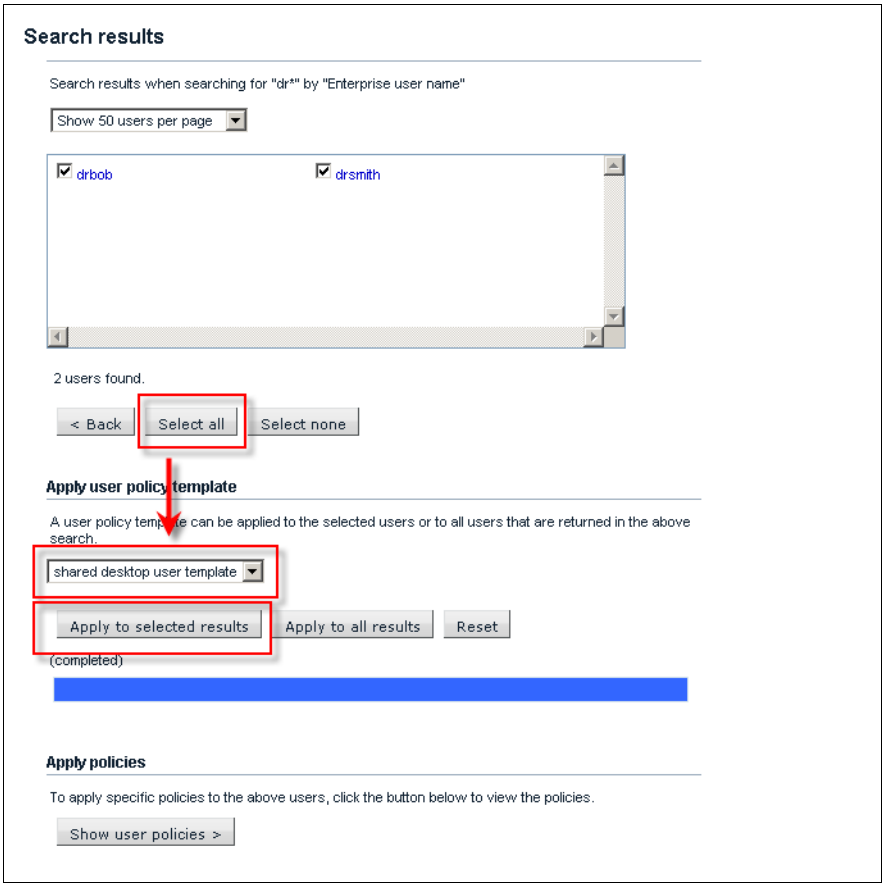

Figure 7-31 Apply shared desktop user template to all doctors as selected users

10. We also change the machine policy template for shared workstations. In AccessAdmin, select **Template assignments**, as shown in Figure 7-32 on page 252, then click the policy template name **Shared workstation, shared desktop, RFID**. Do not check the radio button, but click the actual policy name directly.

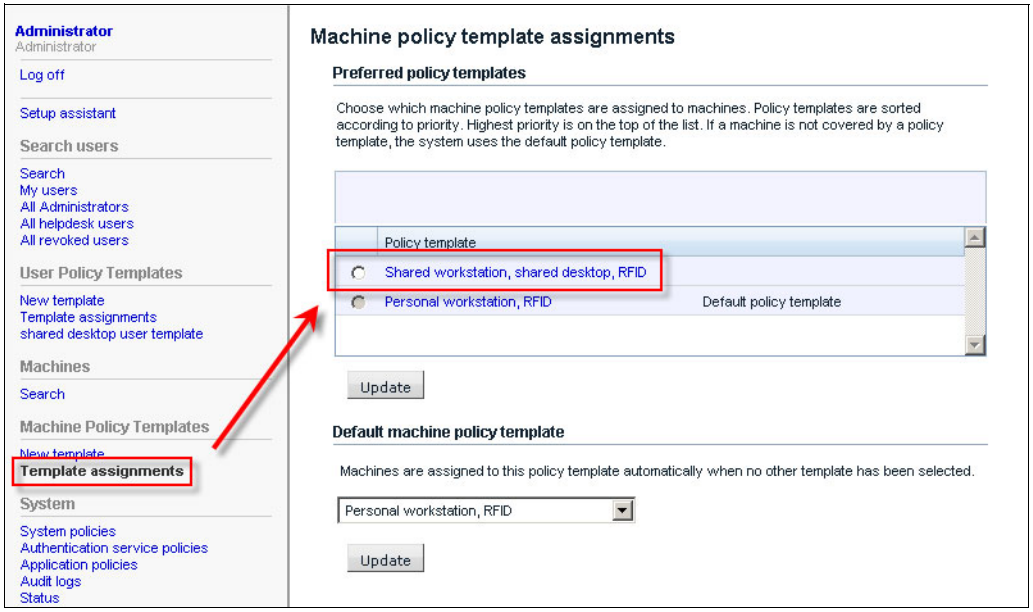

Figure 7-32 Select machine policy template for shared workstations

11. Expand **AccessAgent Policies**, as shown in Figure 7-33 on page 253.
12. Expand **RFID policies**.
13. Change the value of the field “Confirmation countdown duration in seconds, for tapping same RFID on desktop” from 5 seconds to 0 seconds to allow immediate action.
14. Also change the value of the field “Confirmation countdown duration in seconds, for tapping different RFID on desktop” from 5 seconds to 0 seconds to allow immediate action.

Confirmation countdown: This feature offers a grace period during which an unintended authentication action can be canceled. For instance, if someone unintentionally taps their badge to a reader, this feature allows the user time to cancel the following action.

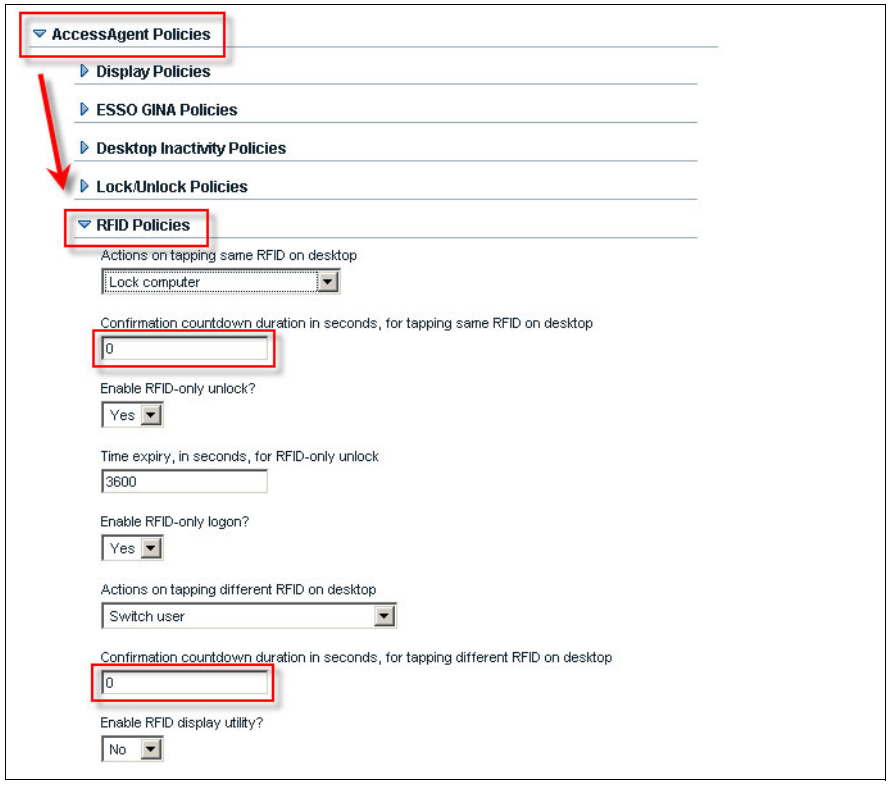

Figure 7-33 Change RFID policy values

15. Scroll to the bottom of the page, and click **Update** to apply the policy changes, as shown in Figure 7-34.

Figure 7-34 Apply policy changes for shared workstation machine policy template

Policy priorities: We configured the same policies on user and machine scope. If the same policy is defined for two scopes (machine and user, or system and user, or machine and system), you must set a priority in case the defined value differs for the two scopes. Look at **IBM Security Access Manager for Enterprise Single Sign-On Policies Definition Guide → Viewing and setting policy priorities** to learn how to use the command-line tool **`managePolPriority.bat`** or **`managePolPriority.sh`** to view and set policy priorities.

The configuration of the policies is complete.

7.2 Using RFID

After we configure the policies, we describe how to log on to a personal desktop with an RFID card:

1. After AccessAgent is installed on a personal workstation and the machine starts, the user sees the IBM Security Access Manager for Enterprise Single Sign-On welcome window, as shown in Figure 7-35, instead of the Windows welcome window.

Figure 7-35 Welcome window

2. The cardio healthcare company employee, `Dr Bob`, exists in the Active Directory. He taps his RFID card on the card reader to log on for the first time after AccessAgent is deployed on his personal workstation. He is prompted for his Windows User name and Password, as shown in Figure 7-36 on page 255. Type the user name and password, and click **OK**.

Figure 7-36 Logon window after tapping the RFID card

3. The user is prompted to provide an answer for the first secret question, by default, “What’s your favorite color?”, as shown in Figure 7-37. Type an answer and click **Next**.

Secret questions: For a more detailed description of the role of these questions, see 6.3.1, “Setting up the self-service questions” on page 201.

Figure 7-37 Type the answer for the first secret question

4. Then, the user is prompted to provide an answer for the second secret question, as shown in Figure 7-38 on page 256, “What’s your favorite fruit?” Type an answer, and click **Next**.

Figure 7-38 Define an answer for the second secret question

5. You can define your own set of questions at the sign up in the AccessAdmin under **System policies** → **Sign Up Policies**, as shown in Figure 7-39 on page 257.

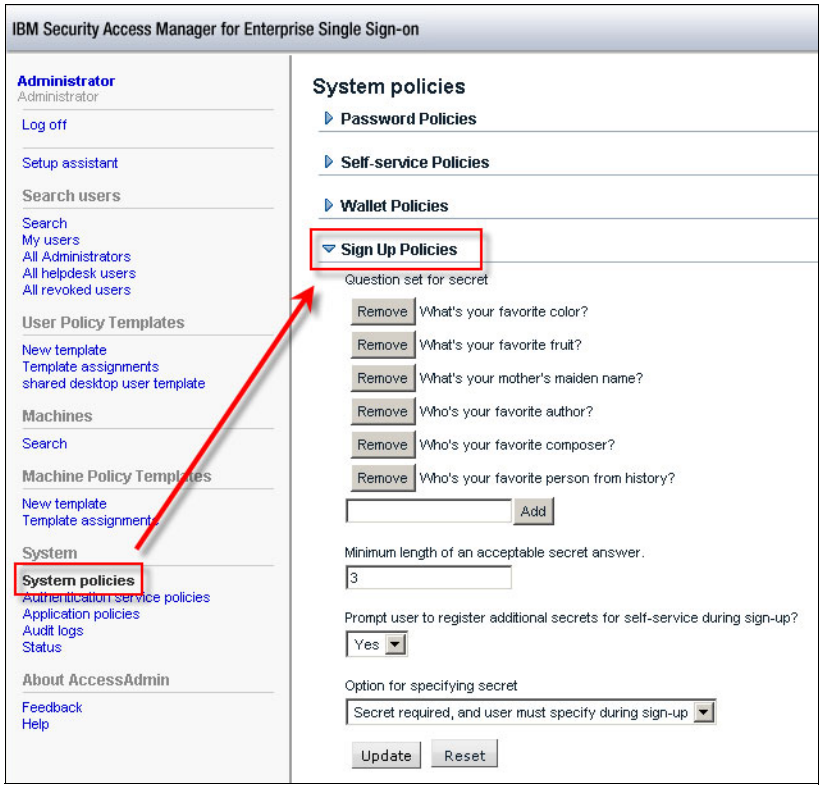

Figure 7-39 Configure Sign Up Policies

6. Now, the Windows logon window is shown (Figure 7-40). The AccessProfile for Windows logon automatically injects the credentials that the user provided in step 2 on page 254. Click **OK** to continue.

Figure 7-40 AccessProfile injects the Windows credentials automatically

7. Right-click the **IBM Security Access Manager for Enterprise Single Sign-On AccessAgent** icon in the icon tray. Select **Manage Wallet** to open the Wallet Manager, as shown in Figure 7-41. The Wallet Manager includes a line with the authentication service cardio.example.com. This line is the directory that we configured previously. You can either change the password entry from Always to Automatic logon here for `Dr Bob` only, or change the default value in the AccessStudio.

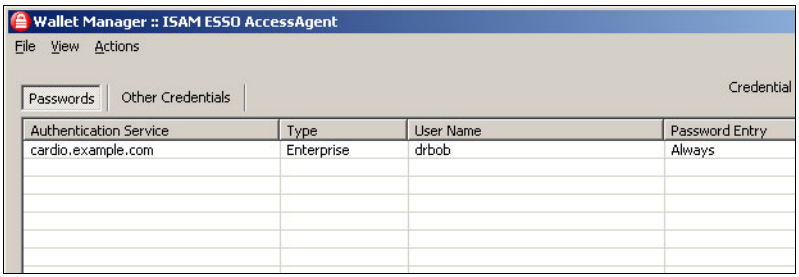

Figure 7-41 Wallet Manager includes Windows credentials for Dr Bob

In the future, AccessAgent always logs on `Dr Bob` automatically to his personal desktop when he taps his RFID card on his computer. `Dr Bob` must also provide his user name and password at the first logon of the day.

The setup and use of strong authentication with IBM Security Access Manager for Enterprise Single Sign-On, combined with RFID cards as a second-factor authentication method, are complete.

We describe the use of shared desktops in combination with virtual desktops in the next chapter.

8

Roaming desktop implementation

In this chapter, we describe a roaming desktop environment. Users at the cardio healthcare company can access their own virtual desktops and the applications that run on those virtual desktops seamlessly from any shared machine.

We integrate IBM Security Access Manager for Enterprise Single Sign-On with VMware Virtual Desktop Infrastructure (VDI) to achieve an integrated end-to-end experience from a user perspective.

We show how to configure the required VMware VDI and IBM Security Access Manager for Enterprise Single Sign-On parameters so that the users that authenticate at a shared workstation (either by entering a password or tapping their RFID badge) can start their business applications in a remote virtual desktop and still have their authentication credentials injected for them. (This approach can also be implemented with finger biometrics.)

The IBM Security Access Manager for Enterprise Single Sign-On Wallet that runs in the Virtual Desktop environment can automatically be synchronized with the Wallet that runs on the shared workstation over a secure virtual channel.

8.1 Cardio healthcare requirements

The medical staff at the hospitals of the cardio healthcare company want to spend nearly all of their time in providing care to their patients. The company wants to reduce the time required of its people to authenticate to IT systems.

Because of the confidential nature of the electronic medical records that the clinicians must update in between caring for their patients, the authentication process needs to be secure. Only correctly authenticated users can be considered for authorized access to this data.

For these reasons, the cardio healthcare company wants to enable its staff to use their individual RFID access badges for logging on to IT systems. The RFID badge must enable the users to use a single sign-on to systems and applications.

From a security perspective, the cardio healthcare company requires that the login process is secure, and that authentication activity for each user is logged for audit and compliance purposes. The integration with IBM Security Access Manager for Enterprise Single Sign-On can meet all of these requirements.

The cardio healthcare company lists the following functional requirements:

- When a user logs on to a shared workstation in a semi-public area by using either a password or an RFID badge (as configured in Chapter 7, “Strong authentication using RFID” on page 227), a connection to this user’s Virtual Desktop is automatically started. The process of logging on the user to this Virtual Desktop must occur through secure and tamperproof methods.
- A user that is logged on to a Virtual Desktop must be able to use the applications already configured in Chapter 5, “Base installation and configuration” on page 107 without having to provide authentication credentials. This function must work the same as if those applications run on the shared workstation from where they connect.
- When a user logs off from a shared workstation, the roaming virtual desktop and its applications must continue to run on the Virtualinfrastructure.
- Shared workstation inactivity policies must be as strict as possible to prevent other people from accessing a lingering Virtual Desktop session. Inactive sessions need to terminate automatically.

8.2 Overview of the roaming desktop features

IBM Security Access Manager for Enterprise Single Sign-On allows a seamless integration into a VMware Virtual Desktop Infrastructure, supporting the required security context for user logon. In this section, we show the component architecture of the solution and explain the logon workflow. First, we explain how the logon workflow works without IBM Security Access Manager for Enterprise Single Sign-On integration. The next section then covers the cardio healthcare company implementation and the configuration details that are required to set it up.

8.2.1 Component architecture overview

The diagram in Figure 8-1 on page 262 shows the targeted solution component architecture. The medical staff members use distributed workstations to automatically log on using their RFID badges and connect to their virtual desktops that are hosted on a VMware ESXi Server.

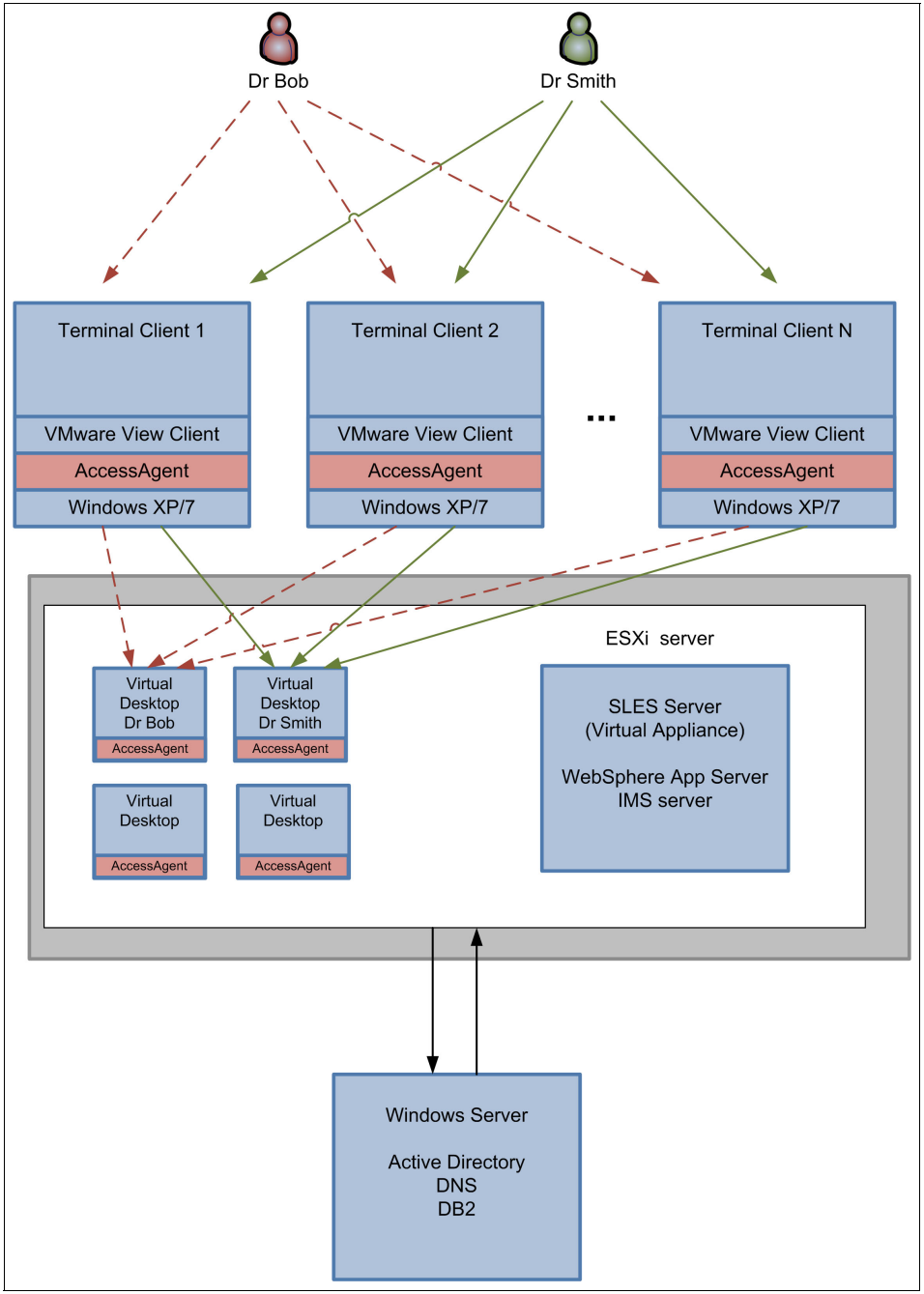

Figure 8-1 Component architecture

Before we describe the implementation for the cardio healthcare company in 8.3, “Cardio healthcare implementation” on page 265, we look at the manual logon process for the VMware virtual desktop.

8.2.2 Logging on manually to the VMware virtual desktop

When a user logs on to a Windows terminal that supports roaming desktops, the user can start the **VMware View Client**. Double-click the icon on the physical desktop to get access to a roaming desktop, as shown in Figure 8-2. This task starts a three-step logon process:

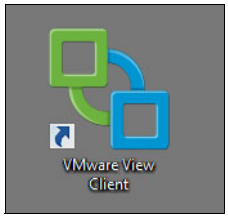

Figure 8-2 VMware View Client icon on the desktop

1. The cardio healthcare user then sees the first VMware Virtual Desktop Infrastructure logon window. The user is prompted for the VMware View Connection Server host name or IP address. The connection server is a broker to the virtual desktops. The cardio healthcare company uses **cardioVCS** as host name for the VMware View Connection Server, as shown in Figure 8-3. Click the host name, and click **Connect**.

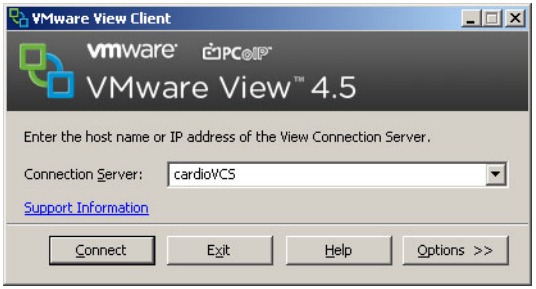

Figure 8-3 First VMware Virtual Desktop Infrastructure logon window

2. The second VMware Virtual Desktop Infrastructure logon window asks for user credentials and the Windows domain. Enter the user name and password. Select a domain. In the example in Figure 8-4 on page 264, we type user name `drbob`, type the password, and select the domain `CARDIO`. When finished, click `Login`.

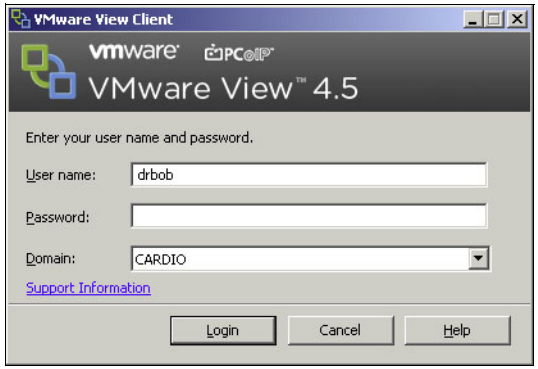

Figure 8-4 Second VMware Virtual Desktop Infrastructure logon window

3. The third VMware Virtual Desktop Infrastructure logon window prompts the user to select from a list of virtual desktops. In the example shown in Figure 8-5, we select Dr. Bob's virtual desktop **VD-drbob**. Change the Display type from the default value `Window - small` to **Full Screen**. Click **Connect** to log on to the virtual desktop.

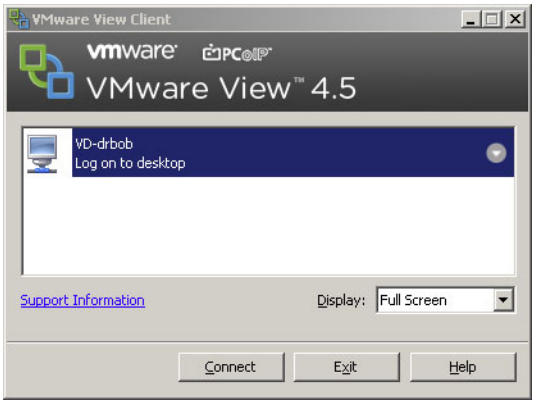

Figure 8-5 Third VMware Virtual Desktop Infrastructure logon window

This three-step logon approach can be optimized by prompting for the user credentials. IBM Security Access Manager for Enterprise Single Sign-On injects the user name and password on behalf of the user and in the correct security context in the product. The VMware View client supports a command-line tool to pass the other missing parameters directly to the client. The cardio healthcare company uses the following command line for this function:

```
"C:\Program Files\VMware\VMware View\Client\bin\wswc.exe" -serverURL
cardioVCS -domainName cardio -desktopLayout fullscreen -desktopName
VD-drbob
```

In this example, `wswc.exe` is the command-line version of the VMware View Client and the input parameters are defined:

serverURL Is the URL to the VMware View connection server, as shown in Figure 8-3 on page 263. In this example, it is `cardioVCS`.

domainName Is the Windows domain name. The cardio healthcare company uses the domain name `cardio`.

desktopLayout Specifies in which mode the virtual desktop must start. Supported values are `fullscreen`, `multimonitor`, `windowLarge`, or `windowSmall`. In this example, we use `fullscreen` for a seamless desktop integration.

desktopName Is the virtual desktop that must automatically be started. In our example, this value is `VD-drbob`. This parameter depends on the user who is logged on to the physical client.

8.3 Cardio healthcare implementation

The VMware Virtual Desktop Infrastructure (VDI) uses the golden master image that includes the preinstalled AccessAgent.

Golden master image: All virtual desktops can be derived from one *golden master image*, which is relevant for volatile images that are provisioned and de-provisioned on demand.

VMware View clones this golden master image, and users log on to these clones. It is important that AccessAgent is installed successfully on the master image and that the certificate is installed by setting the IMS Server location; click **Start** → **All Programs** → **ISAM ESSO AccessAgent** → **Set IMS Server Location**.

VMware Virtual Desktop Infrastructure supports two types of virtual desktops:

- A volatile image is provisioned and de-provisioned on demand, every time that a user must connect to a virtual desktop. The cardio healthcare company does not use this model, because the company wants to allow users to run applications continuously in their virtual environment, even after the users log off.
- A persistent image is provisioned for a specific user and stays resident in the background on the VMware ESX server even if the user disconnects. This scenario allows the cardio healthcare employees to keep their personal virtual

desktop and their applications up and running, regardless from which Windows terminal client they access their virtual desktop. This approach saves time, because the employees do not need to start their applications every time they log on.

The cardio healthcare company uses a global naming convention for its computers. The Windows terminals (shared physical machines), from which users log on to their virtual desktops, have host names that start with `WT`, for example, `WT1234`. The virtual desktops, to which employees connect from the different terminals, start with `VD` and include the Windows user ID in their host name, for example `VD-drbob`.

The organization automates the user logon to the virtual desktops within the IBM Security Access Manager for Enterprise Single Sign-On user policy for all users. In the following steps, we show the configuration for the automatic start of the personal (persistent) virtual desktop for the currently logged on user. The script is only executed if the user logs on to a Windows terminal client that serves as the VMware Virtual Desktop Infrastructure client for the user (host name starts with `VD-`). Follow these steps:

1. Log on to AccessAdmin, for example, by using:

   `https://imsva/admin/faces/auth/login.xhtml`

2. Under User Policy Templates on the left pane, select the **Default user template** (see Figure 8-6 on page 267).
3. Expand **AccessAgent Policies**, and expand **Logon/Logoff Policies** (see Figure 8-6 on page 267).

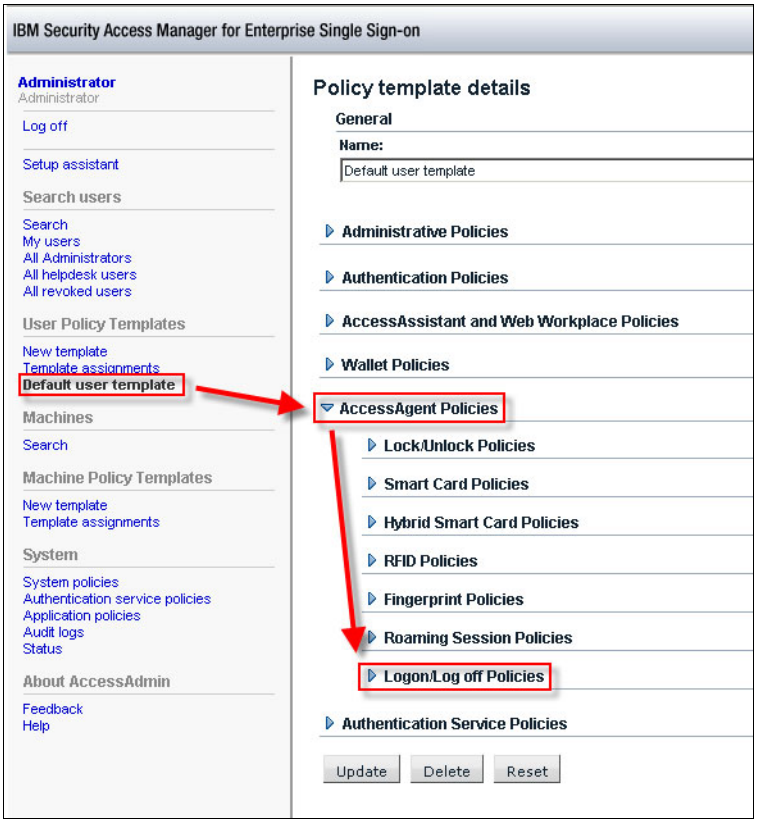

Figure 8-6 Selecting AccessAgent Logon/Logoff Policies

4. As shown in Figure 8-7 on page 268, change **“**Enable logon script during user logon?” to **Yes**. Change the Logon script type to **VBScript**. And, add the following VBScript code in the Logon script code box:

```
Set WshNetwork = WScript.CreateObject("WScript.Network")
If Left(WshNetwork.ComputerName, 2)="WT" Then
   strVD = "VD-" & (wshNetwork.UserName)
   strExec = """c:\Program Files\VMware\VMware"
   strExec = strExec & " View\Client\bin\wswc.exe"""
   strExec = strExec & " -serverURL cardioVCS"
   strExec = strExec & " -domainName cardio"
   strExec = strExec & " -desktopLayout fullscreen"
   strExec = strExec & " -desktopName " & strVD
   set objShell = createobject("Wscript.Shell")
   objShell.Run strExec
End If
```

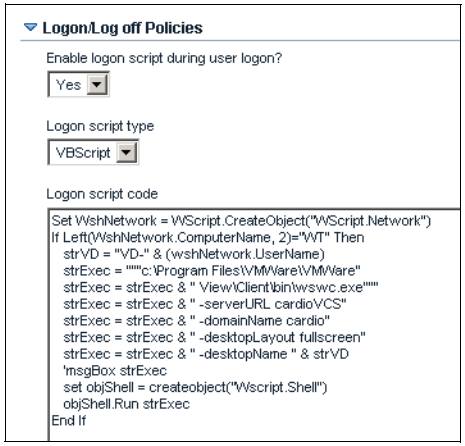

Figure 8-7 Enable and configure the Logon policy

5. Scroll to the end of the page, and click **Update**.

> **Readability:** The variable `strExec` in the VBScript is broken into several lines in this example for better readability.

An alternative approach is to start the VMware View Client only if the client binaries (`wswc.exe`) are on the client.

8.3.1 Usage scenarios

In this section, we describe the usage scenarios for users that use the same shared workstation to access their personal virtual desktops.

Logon scenario

This scenario describes the logon process to a VMware virtual desktop through a shared workstation. For a graphical representation, see Figure 8-1 on page 262. Follow these steps:

1. Terminal client 1 starts, for example, in the morning. The generic terminal client user defined for shared desktops is logged on automatically.
2. IBM Security Access Manager for Enterprise Single Sign-On automatically locks the screen.
3. Dr. Bob logs on to terminal client 1 by tapping his card on the RFID card reader.

4. Dr. Bob is logged on to Windows as the generic terminal client user, but he is logged on to his personal Wallet.

   Password: If this use is the first logon of the day, Dr. Bob also must provide his password for AccessAgent at the Windows logon.

5. The cardio healthcare company defined a logon script in the IBM Security Access Manager for Enterprise Single Sign-On user policy template for shared users. This policy also applies to Dr. Bob. The Visual Basic script automatically starts the VMware View Client from the command line by using `wswc.exe`.
6. The command line provides the parameters for the VMware connection server to avoid VMware View Client prompting the user for this parameter.
7. The VMware View GUI prompts for the user name and password. An AccessProfile for `wswc.exe` automatically injects the credentials for Dr. Bob.
8. Dr. Bob's virtual desktop that runs on the ESX server starts in `fullscreen` mode.

Whenever Dr. Bob taps his RFID access badge at any shared physical workstation in one of the hospitals, he is automatically logged on to that workstation. And, he is seamlessly routed through his own Virtual Desktop that runs on a remote server that contains all of his applications. He can start these applications by using the IBM Security Access Manager for Enterprise Single Sign-On technology as though the applications run on the physical workstation where he logged on.

Lock screen scenario

In this scenario, we describe the steps when a cardio healthcare user locks the screen of a shared workstation. Follow these steps:

1. Dr. Bob taps his RFID card on terminal client 1, on which he connects to his virtual desktop.
2. The user policy template for Dr. Bob defined a screen lock script, which terminates the VMware View Client.
3. The cardio healthcare company defined a screen lock script in the IBM Security Access Manager for Enterprise Single Sign-On user policy template for shared users. This policy also applies to Dr. Bob. The Visual Basic script terminates the VMware View client session for Dr. Bob.
4. AccessAgent locks the screen.
5. The virtual desktop continues to run on the ESX server.

Unlock screen scenario

In this scenario, we describe the steps when a user unlocks the screen of a shared workstation:

1. Dr. Bob taps his RFID card on terminal client 1, on which he connects to his virtual desktop.
2. The same logon workflow, as described in “Logon scenario” on page 268, applies.

Another user in the running terminal client scenario

In this scenario, we describe the workflow when a second user (Dr. Smith) logs on to a shared workstation while a user (Dr. Bob) is already logged on:

1. Dr. Smith logs on to terminal client 1 by tapping her card on the RFID card reader.
2. The AccessAgent logoff script for Dr. Bob terminates VMware View Client.
3. Dr. Bob is logged off from the workstation.
4. Dr. Smith is logged on instantly.
5. The same logon workflow, as described in “Logon scenario” on page 268, applies for Dr. Smith.

8.4 Conclusion

IBM Security Access Manager for Enterprise Single Sign-On offers a fully automated logon and logoff with the VMware Virtual Desktop Infrastructure virtual desktops. If a user logs in to a Windows machine, AccessAgent executes the user logon policy automatically. The VMware View client is started by the policy only if the user logged in to a shared workstation (often in the semi-public areas of the hospitals) that functions as a client to the user’s personal virtual desktop. When the cardio healthcare company user logs off from the physical machine, the virtual desktop stays in its current state. When the user then connects to another terminal client, the user connects to that same personal virtual desktop again. This concept is the roaming desktop.

9

Implementing operational requirements

In most organizations, operational activities start immediately after the deployment of new technology, such as IBM Security Access Manager for Enterprise Single Sign-On. In this chapter, we describe the required operational activities of the cardio healthcare company environment.

We describe how to update the IBM Security Access Manager for Enterprise Single Sign-On infrastructure with fix packs. We explain how managing the audit logs in the database can help improve system performance.

We explain the backup and restore procedures for IBM Security Access Manager for Enterprise Single Sign-On components.

We describe how to use the IBM Security Access Manager for Enterprise Single Sign-On reporting feature (Tivoli Common Reporting) to fulfill the compliance needs of the cardio healthcare company.

9.1 Fixes

Fix packs are the periodic bundling of interim fixes and other resolved *Authorized Program Analysis Reports* (APARs). These fix packs provide changes to the software that can resolve known problems, add new functions, and help the software operate efficiently. Administrators must regularly ensure that fix packs are deployed to keep the system up-to-date. Administrators must be proactive and plan for, test, and implement fix packs as they are released. They must not wait until a known issue arises. We illustrate the steps.

APAR: An Authorized Program Analysis Report (APAR) is a formal report to IBM development of a problem caused by a suspected defect in a current release of an IBM program. If IBM development is able to confirm the existence of the defect, IBM updates the APAR with any known work-around. IBM might indicate which future release, if any, of the IBM program they intend to target for a formal fix to the defect and whether a Program Temporary Fix (PTF) is planned. The APAR is published so that it is visible to supported clients.

9.1.1 Finding fix levels

Administrators can follow these steps to identify the fix levels of AccessAgent and the IMS Server:

- AccessAgent

  The AccessAgent version and the fix level can be located by opening AccessAgent after a successful logon, as shown in Figure 9-1.

Figure 9-1 AccessAgent version

- IMS Server

  To locate the IMS Server version, you need to log in to AccessAdmin with administrative privileges, and click **Status** in the System category, as shown in Figure 9-2.

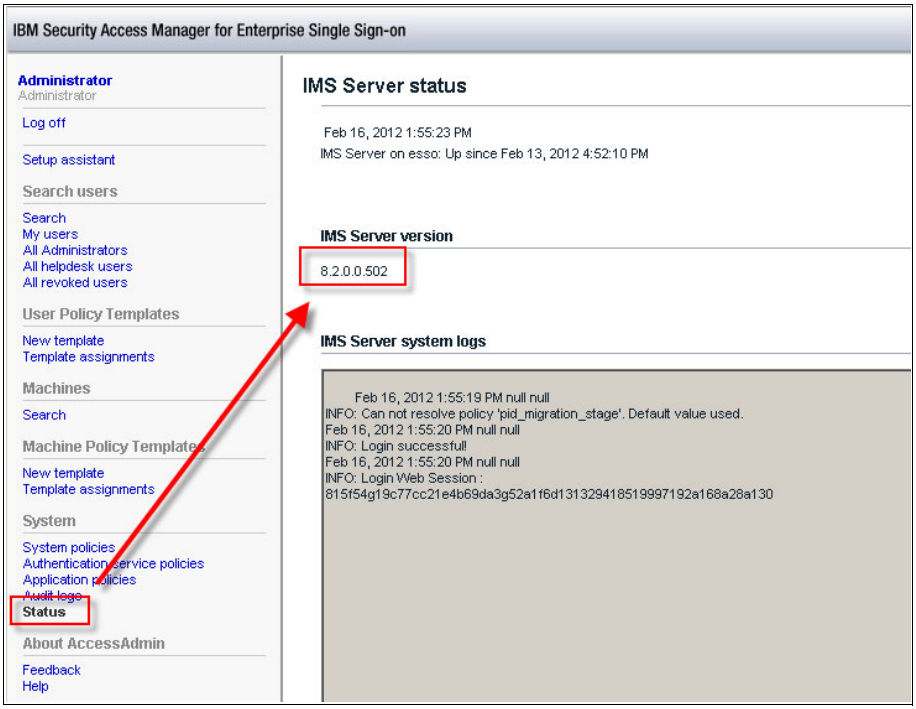

Figure 9-2 IMS Server version

9.1.2 Obtaining fixes

Administrators can determine the available fixes for IBM Security Access Manager for Enterprise Single Sign-On by checking the product support website:

1. Go to the IBM Software Support website for IBM Security Access Manager for Enterprise Single Sign-On:

   `http://www.ibm.com/software/sysmgmt/products/support/index.html`

2. Select IBM Security Access Manager for Enterprise Single Sign-On from the Support for specific Tivoli products drop-down list. A list of the most recent fixes shows in the Download section of the page.

3. Click the name of a fix to read the description. You can also download the fix.

9.1.3 Receiving fix notifications

Administrators can enable email notifications and subscriptions to receive notifications about any new fix packs and other news about IBM Security Access Manager for Enterprise Single Sign-On. To receive email notifications about IBM Security Access Manager for Enterprise Single Sign-On fixes and other news about IBM Security Access Manager for Enterprise Single Sign-On, follow these steps:

1. Go to the IBM Software Support website for IBM Security Access Manager for Enterprise Single Sign-On:

   `http://www.ibm.com/software/sysmgmt/products/support/index.html`
2. Select **IBM Security Access Manager for Enterprise Single Sign-On** from the Support for specific Tivoli products drop-down list.
3. Click **My Support** in the upper-right corner of the page. A sign-in page is displayed.
4. If you are already registered, go to the next step. If you are not registered, click **Register now** to establish your user ID and password.
5. Sign in to My support.
6. Click the **Edit profile** tab.
7. Select **Software** → **Security** → **Access** in the fields that are displayed.
8. Select **IBM Security Access Manager for Enterprise Single Sign-On** from the list of products displayed.
9. Click **Add products**.
10. To enable email notification, click **Subscribe to email** at the top of the page.
11. From the list, click **Software**.
12. Select the check boxes that best describe the email notifications that you want to receive.
13. Click **Update**.
14. Sign out of the session by clicking **Sign out** or click **Go to my personalized page** to see your personalized support page.

9.2 Audit log maintenance

IBM Security Access Manager for Enterprise Single Sign-On stores audit log and historical information in the database. For better performance, periodically

administrators need to prune or remove audit logs and historical information from the database and reduce the database size.

To download readily available database scripts for audit log maintenance, see this website:

http://www.ibm.com/support/docview.wss?uid=swg21572941

9.3 Database maintenance

IBM Security Access Manager for Enterprise Single Sign-On stores user and Wallet information in the database, and a plan for routine database maintenance tasks to optimize database performance must be available. You must establish routine database maintenance procedures to update statistics and reorganize database tables on a regular basis.

Certain database products support *automatic maintenance*, which you need to enable if you do not plan to perform any manual maintenance tasks. To determine the types of database maintenance tasks that you can plan and complete, see the database vendor documentation.

9.4 Cached Wallet maintenance

IBM Security Access Manager for Enterprise Single Sign-On stores cached Wallets in the database. For better performance, administrators must periodically remove expired or duplicated cached Wallets from the database.

To download available database scripts for cached Wallet maintenance, see this website:

http://www.ibm.com/support/docview.wss?uid=swg21572941

9.5 Backup and restore procedures

Backup and restore procedures are necessary to ensure high availability and disaster recovery. Developing plans for backup and recovery procedures can be part of an overall disaster recovery plan. In the backup and restore plan for IBM Security Access Manager for Enterprise Single Sign-On, administrators must include the following components:

- 9.5.1, “WebSphere Application Server profile” on page 276

- 9.5.2, “IMS database” on page 282
- 9.5.3, “IMS Server configuration” on page 287

9.5.1 WebSphere Application Server profile

WebSphere Application Server provides the **`manageprofiles`** command to back up and restore profiles.

Cardio healthcare company administrators can follow these steps to back up the WebSphere Application Server Profile:

1. Log in to the virtual appliance with the default virtual image user ID that you created in step 12 on page 123. The cardio healthcare company created the user ID `virtuser`.
2. Stop the WebSphere Application Server by double-clicking the **Stop the application server (server1)** desktop icon, as shown in Figure 9-3.

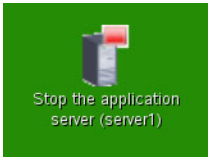

Figure 9-3 Stop the WebSphere Application Server

3. Stop the IBM HTTP Server by double-clicking the **Stop IBM HTTP Server** desktop icon, as shown in Figure 9-4.

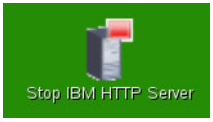

Figure 9-4 Stop the IBM HTTP Server

4. After successfully stopping the WebSphere Application Server and HTTP server, right-click the desktop and select **Open in Terminal**, as shown in Figure 9-5 on page 277.

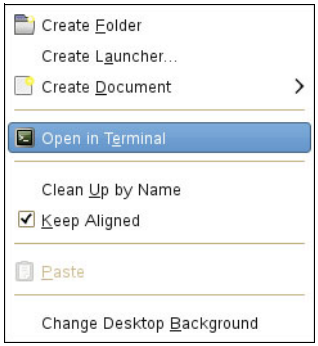

Figure 9-5 Open in Terminal

5. The opened terminal looks similar to Figure 9-6.

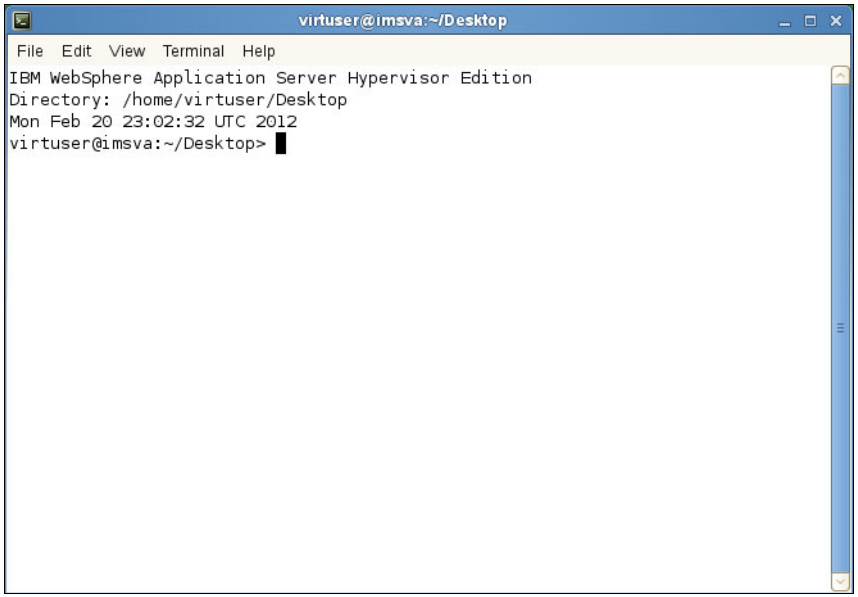

Figure 9-6 Opened command prompt

6. Change the directory to `/opt/IBM/WebSphere/AppServer/bin`, as shown in Figure 9-7 on page 278.

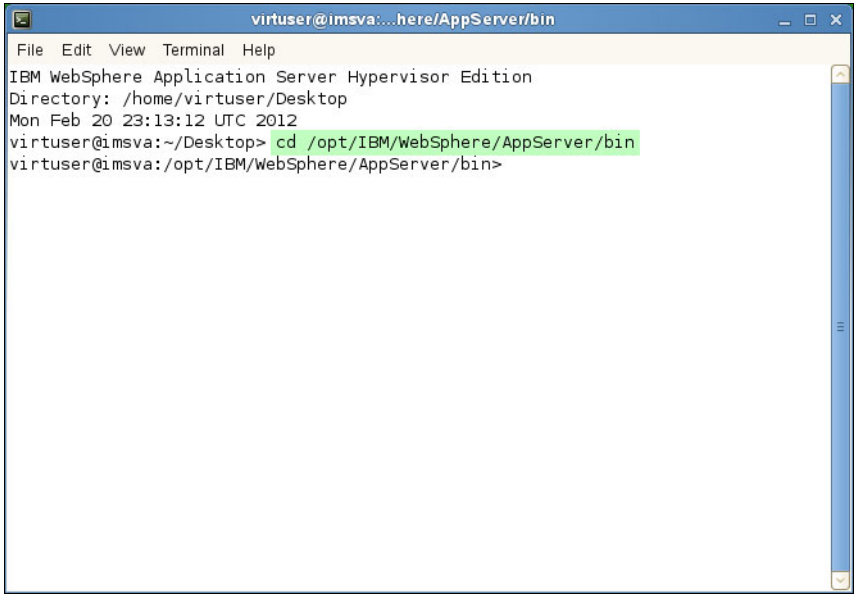

Figure 9-7 Change directory

7. Use the WebSphere Application Server **`manageprofiles`** command with the **`backupProfile`** parameter, as shown in Figure 9-8 on page 279, to back up the profile named `DefaultAppSrv01`.

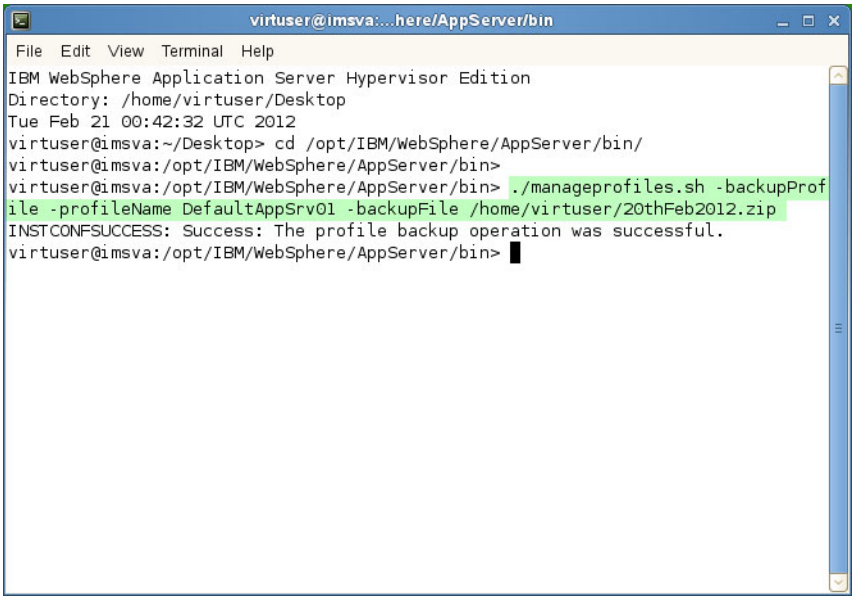

Figure 9-8 Back up the WebSphere Application Server Profile

Cardio healthcare company administrators can use these steps to restore the backup of the WebSphere Application Server Profile:

1. Log in to the virtual appliance with the default virtual image user ID that you created in step 12 on page 123. The cardio healthcare company created the user ID `virtuser`.
2. Stop the WebSphere Application Server by double-clicking the **Stop the application server (server1)** desktop icon, as shown in Figure 9-9.

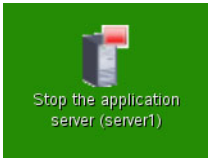

Figure 9-9 Stop the WebSphere Application Server

3. Stop the IBM HTTP Server by double-clicking the **Stop IBM HTTP Server**, as shown in Figure 9-10 on page 280.

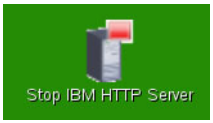

Figure 9-10 Stop IBM HTTP Server

4. Ensure that the `/opt/IBM/WebSphere/profiles` directory does not contain a folder with the same name as the profile `DefaultAppSrv01` to be restored. If a duplicate exists, you can delete the profile with the **`manageprofiles`** command or move the folder to another location.
5. After successfully stopping WebSphere Application Server and HTTP server, right-click the desktop and select **Open in Terminal**, as shown in Figure 9-11.

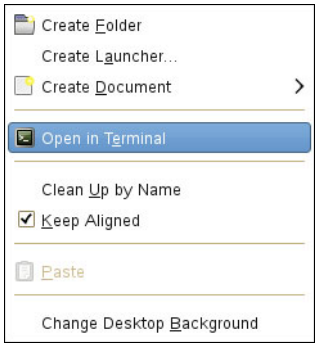

Figure 9-11 Open in Terminal

6. The opened terminal looks similar to Figure 9-12 on page 281.

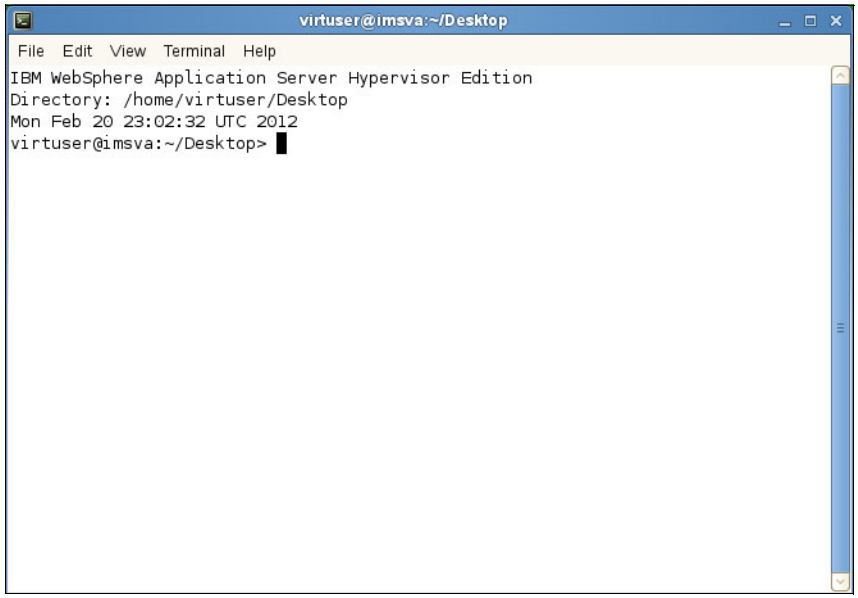

Figure 9-12 Opened command prompt

7. Change the directory to `/opt/IBM/WebSphere/AppServer/bin`, as shown in Figure 9-13.

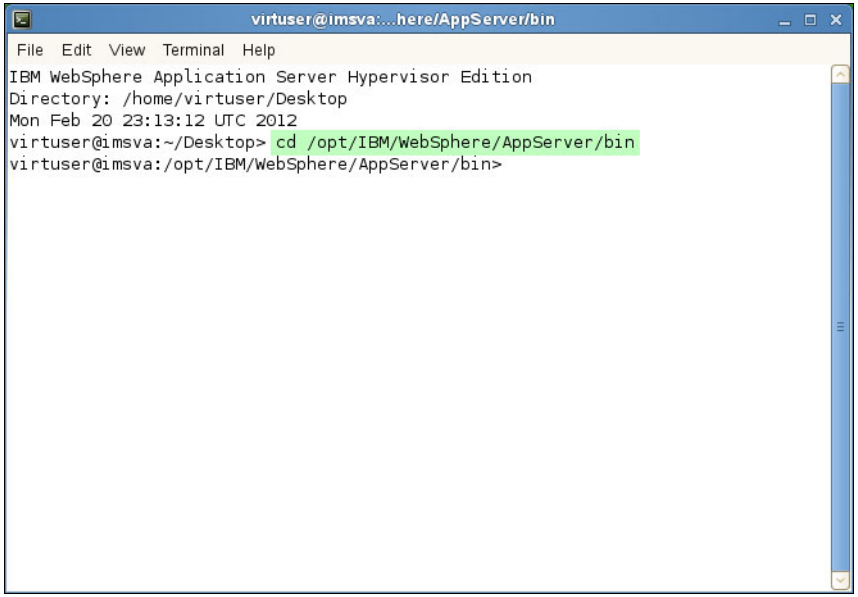

Figure 9-13 Change directory

8. Use the WebSphere Application Server **manageprofiles** command with the **restoreProfile** parameter, as shown in Figure 9-14, to restore the profile named `DefaultAppSrv01`.

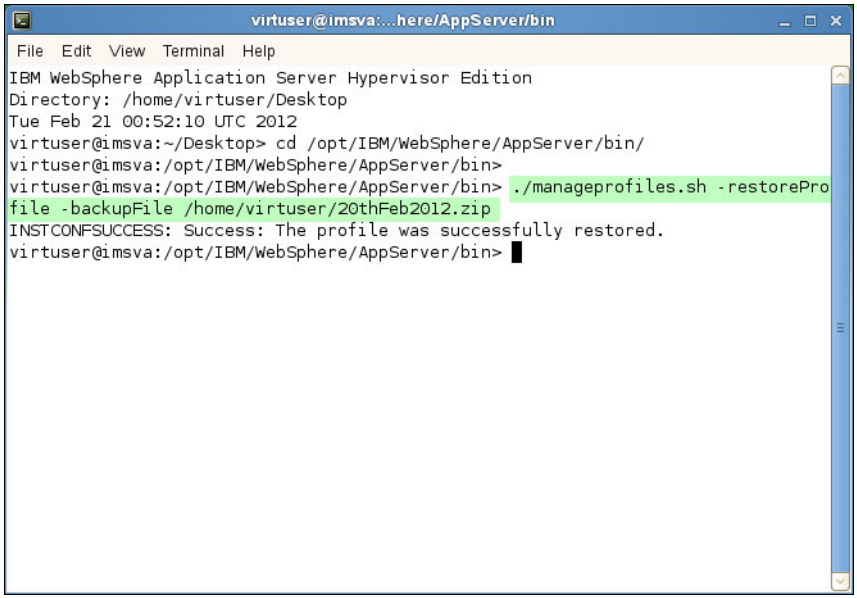

Figure 9-14 Restore WebSphere Application Server Profile

Restore the database first: If you perform a restore task as part of the IBM Security Access Manager for Enterprise Single Sign-On restoration procedure, do not start the profile unless the database is restored.

9.5.2 IMS database

The cardio healthcare company uses IBM DB2 as the IMS database with the IBM Security Access Manager for Enterprise Single Sign-On virtual appliance. IBM DB2 provides the command-line utilities and the GUI interface for backing up the database and restoring the database backup image. In this section, we demonstrate the backup and restore procedures by using command-line utilities.

Administrators can follow these steps to back up the IMS database:

1. Log in as the `db2admin` user to Microsoft Windows Server 2003 where DB2 is installed.
2. Click **Start** and select **Run**, as shown in Figure 9-15 on page 283.

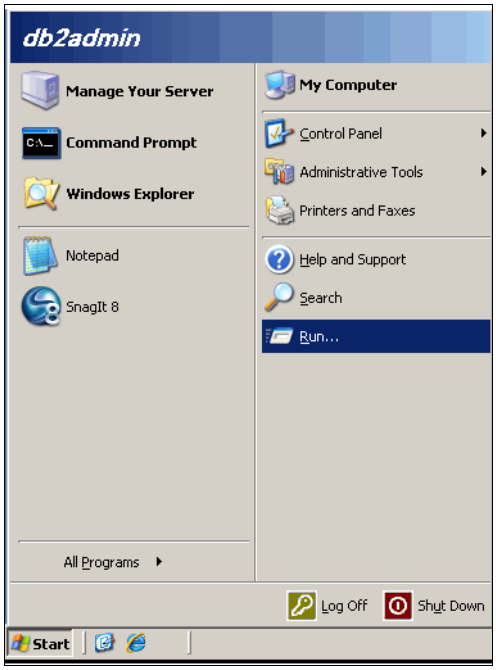

Figure 9-15 Run option from Windows Start

3. Type `cmd`, as shown in Figure 9-16, and click **OK**.

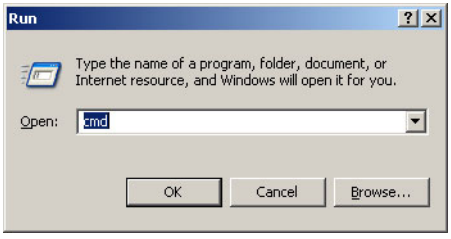

Figure 9-16 Opening command prompt

4. After you click **OK**, the command prompt opens, as shown in Figure 9-17 on page 284.

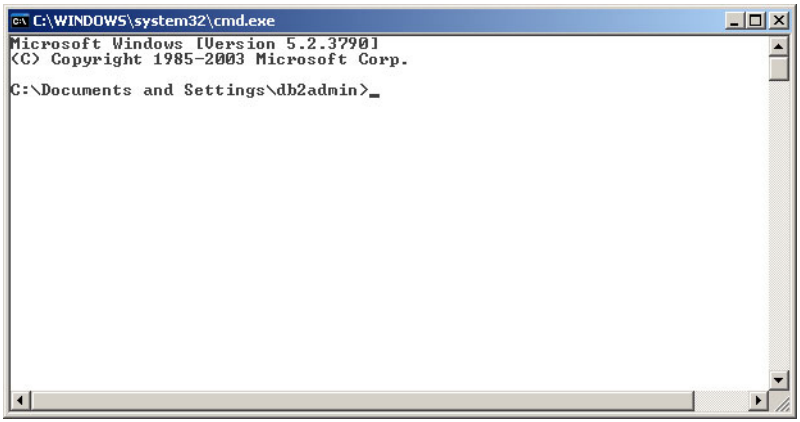

Figure 9-17 Windows command prompt

5. Change the directory to the DB2 installed directory. In the cardio healthcare company, DB2 is installed in `C:\Program Files\IBM\SQLLIB`, as shown in Figure 9-18.

Figure 9-18 Change the directory

6. Execute the **db2cmd** command to start the DB2 command prompt, as shown in Figure 9-19 on page 285.

Figure 9-19 db2cmd command

7. Ensure that the database is not in use by using the **`list applications`** command, as shown in Figure 9-20. If the database is in use, stop all applications that use the `imsvadb` database.

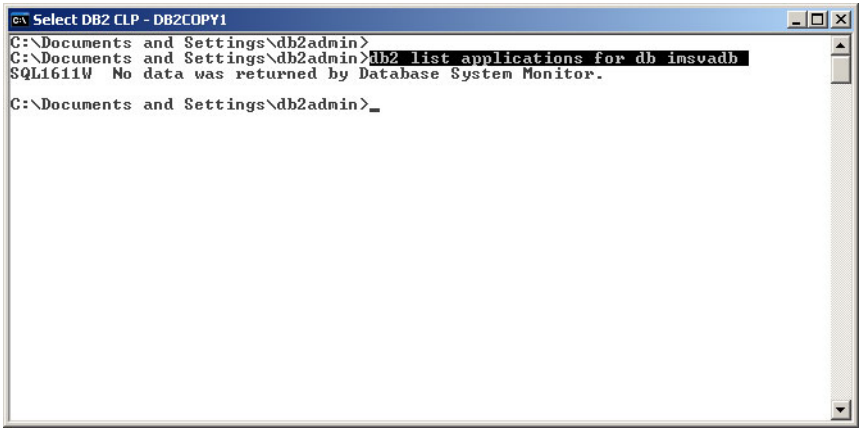

Figure 9-20 db2 list application command

8. Execute the **`db2 backup`** command to back up the `imsvadb` database, as shown in Figure 9-21 on page 286. Also, make a note of the time stamp of the backup image. This time stamp is required while restoring the database backup image.

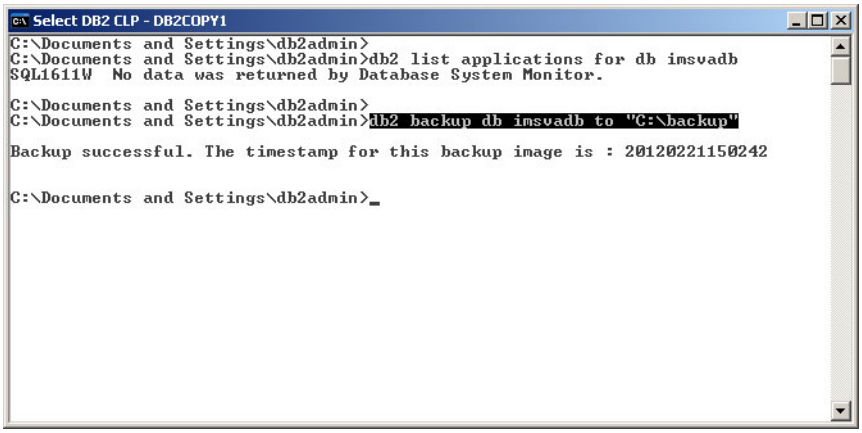

Figure 9-21 imsvadb database backup command

Administrators can use the following steps to restore a backup of the IMS database:

1. Log in as the `db2admin` user to Windows Server 2003 where DB2 is installed.
2. Start another command prompt window.
3. Change the directory to the DB2 installed directory. In the cardio healthcare company, db2 is installed in `C:\Program Files\IBM\SQLLIB`, as shown in Figure 9-22.

Figure 9-22 Change the directory to the db2 installed directory

4. Execute the **`db2cmd`** command-line interpreter.

5. Ensure that the database is not in use by using the **`list applications`** command, as shown in Figure 9-23. If the database is in use, stop all applications that use the `imsvadb` database.

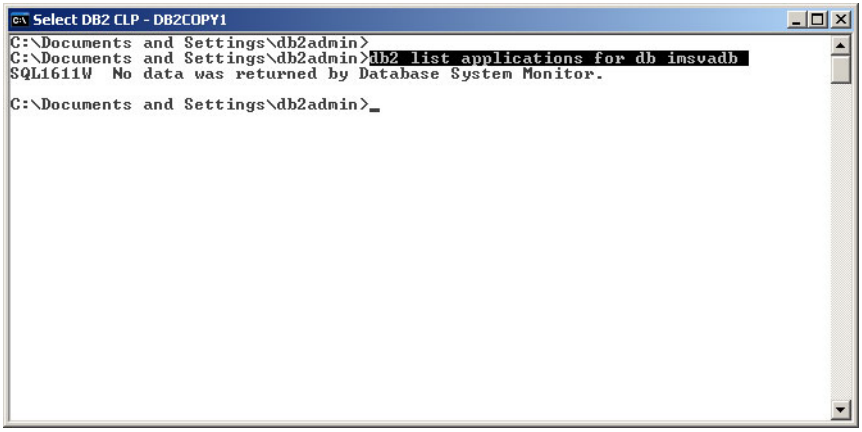

Figure 9-23 db2 list application command

6. Execute the **`db2 restore`** command to restore a backup of the database with the correct time stamp, as shown in Figure 9-24.

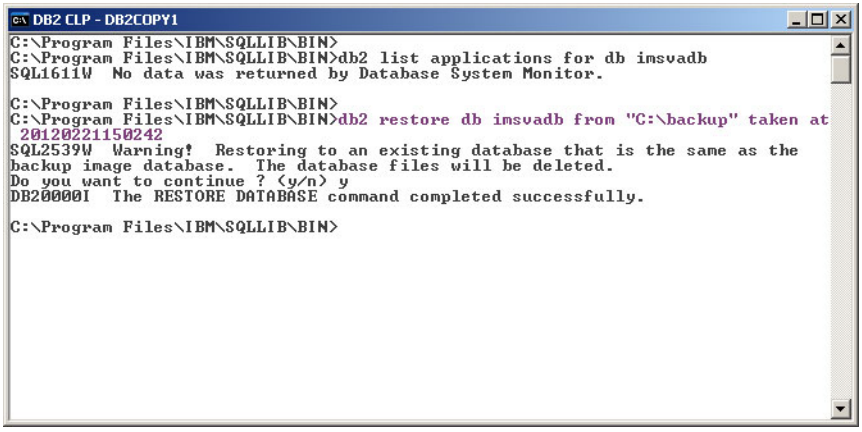

Figure 9-24 Restoring the imsvadb database by using the db2 restore command

9.5.3 IMS Server configuration

The IMS Server application runs on the WebSphere Application Server platform. The IMS Server application provides *export* and *import* configuration tools in the IMS Configuration utility to back up and restore the IMS Server configuration.

Administrators can follow these steps to export the IMS configuration:

1. Open a web browser and enter the following web address:

   `https://esso:9043/webconf`

   If the IMS Server is installed, this address directs you to the login page where you can log in with `virtuser` and the password, as shown in Figure 9-25.

Figure 9-25 IMS Configuration login page

2. On a successful logon, you are presented with the IMS Configuration Utility that includes an Export IMS Configuration option under Utilities, as shown in Figure 9-26 on page 289. Click **Export IMS Configuration** to start the configuration process.

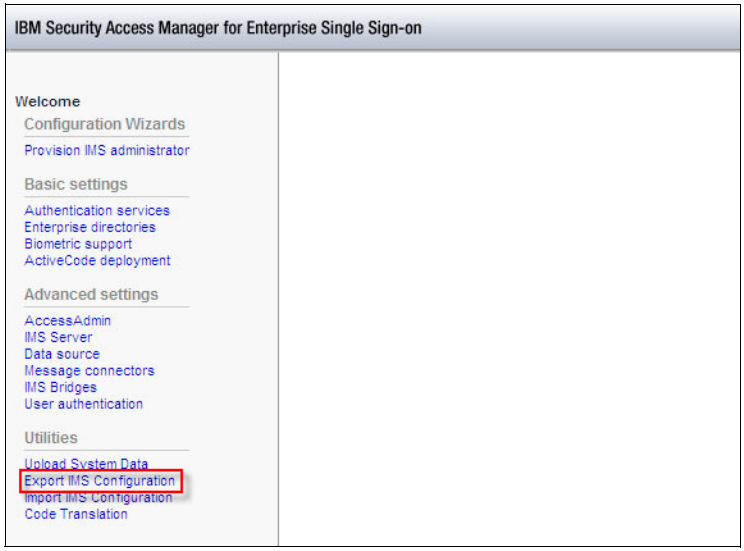

Figure 9-26 IMS Configuration Utility

3. After selecting Export IMS Configuration, you are presented with the Export IMS Server Configuration wizard, as shown in Figure 9-27. Click **Begin**.

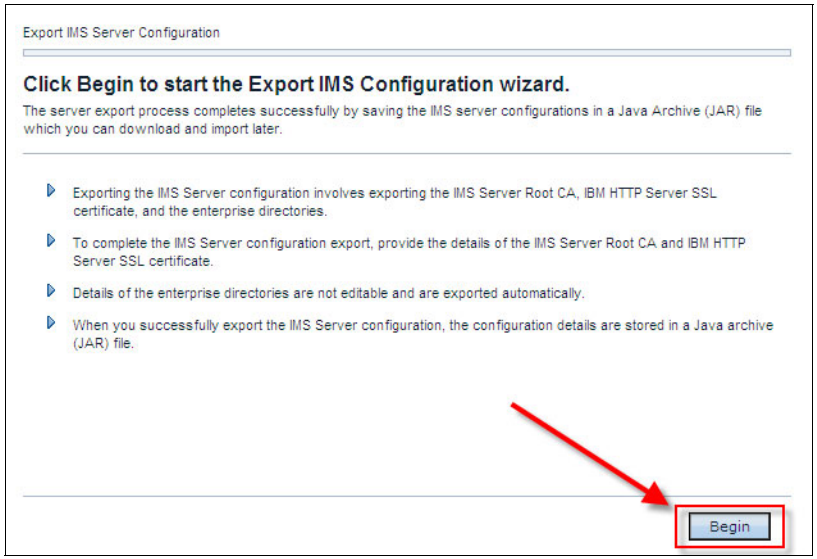

Figure 9-27 Export IMS Configuration wizard

4. Accept the default values in the WebSphere Application Server Root Certificate Authority (CA) form, because the cardio healthcare company used the default values. Click **Next**, as shown in Figure 9-28.

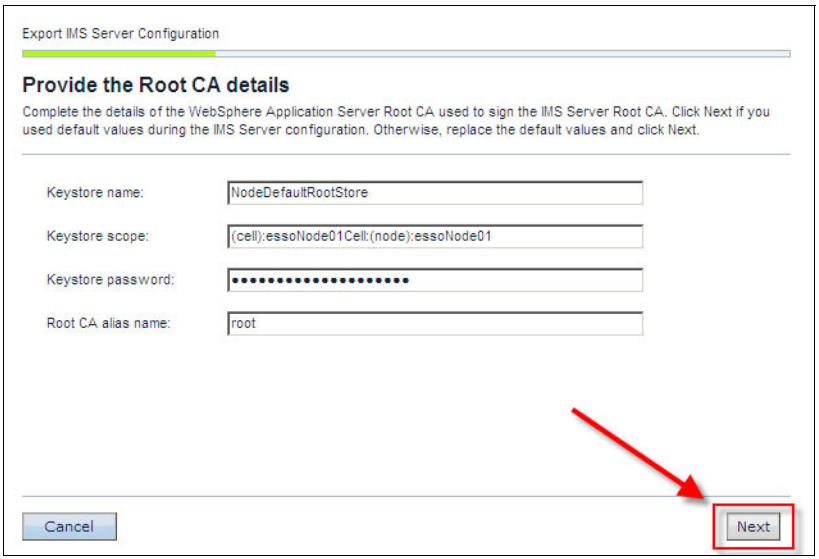

Figure 9-28 WebSphere Application Server Root Certificate Authority form

5. Accept the default values of the IBM HTTP Server Secure Sockets Layer (SSL) Certificate, because the cardio healthcare company used the default values. Click **Next**, as shown in Figure 9-29 on page 291.

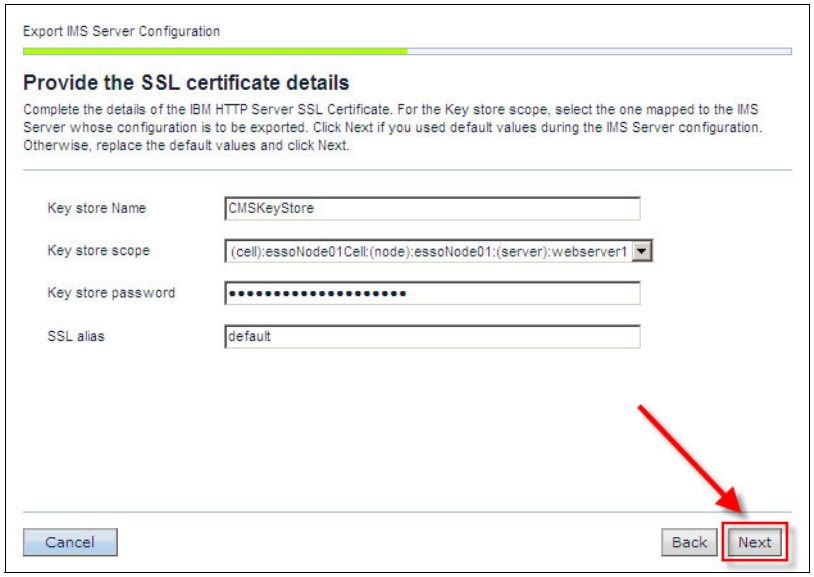

Figure 9-29 IBM HTTP Server SSL Certificate form

6. Review the summary of the IMS Server Configuration to be exported, and click **Export**, as shown in Figure 9-30.

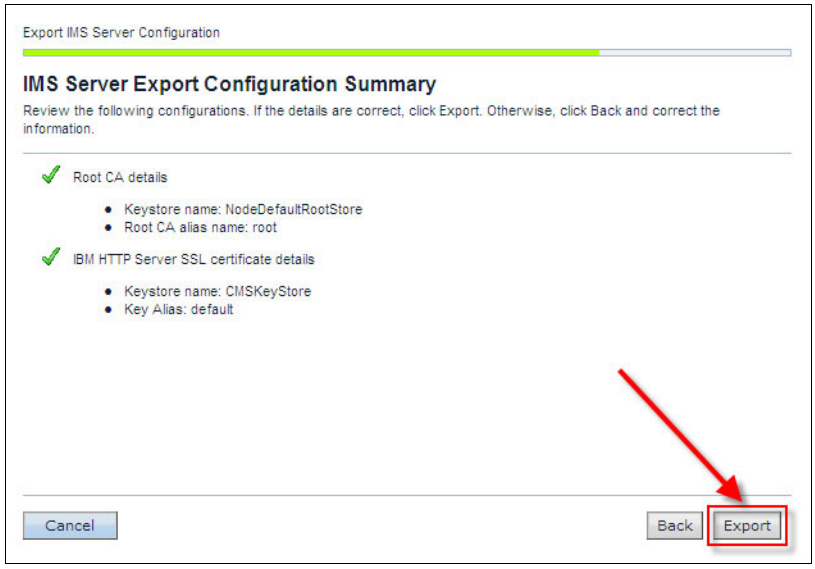

Figure 9-30 IMS Server Export Configuration Summary

7. On a successful IMS Server Configuration export, you are presented with a download option, as shown in Figure 9-31. Click **Download** to download the IMS Server configuration.

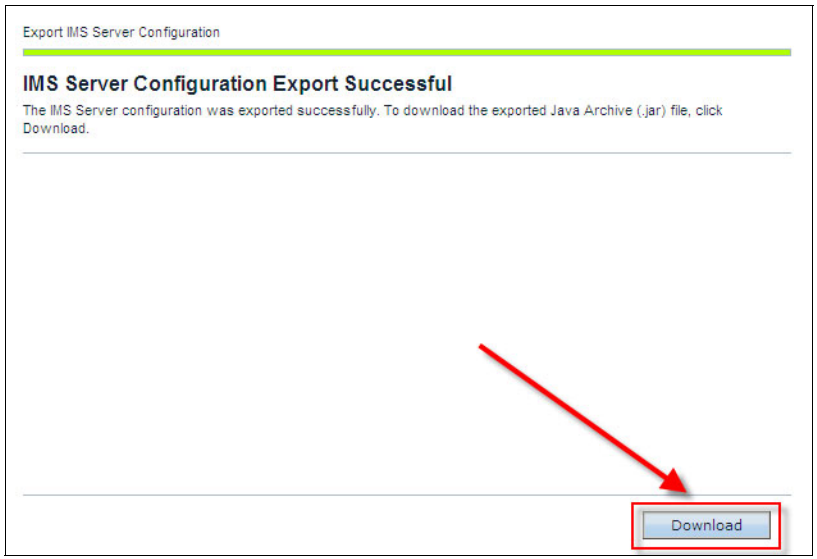

Figure 9-31 Successful IMS Server configuration export

Important: The export configuration tool does not back up the IMS Server database and manual changes to the WebSphere Application Server Profile.

Administrators can follow these steps to import the IMS Server configuration:

1. Open a web browser and enter the following web address:

   `https://esso:9043/webconf`

   If the IMS Server is installed, this address directs you to the login page where you need to log in with `virtuser` and the password, as shown in Figure 9-32 on page 293.

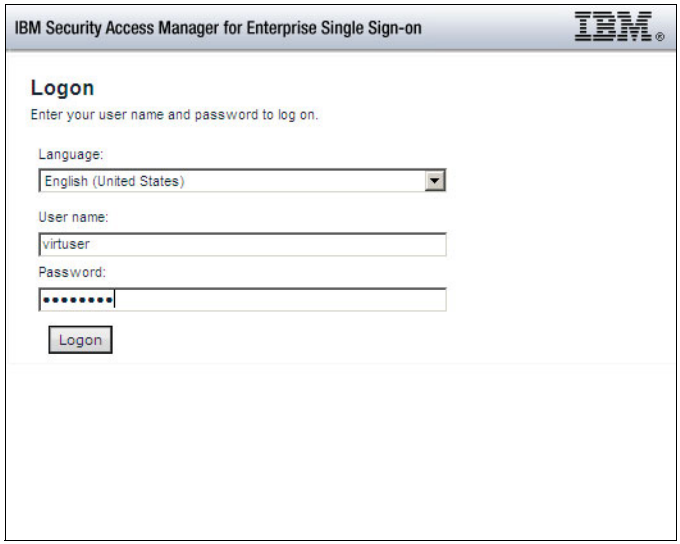

Figure 9-32 IMS Configuration login page

2. After a successful logon, you are presented with the IMS Configuration Utility. Under Utilities, select **Import IMS Configuration**, as shown in Figure 9-33.

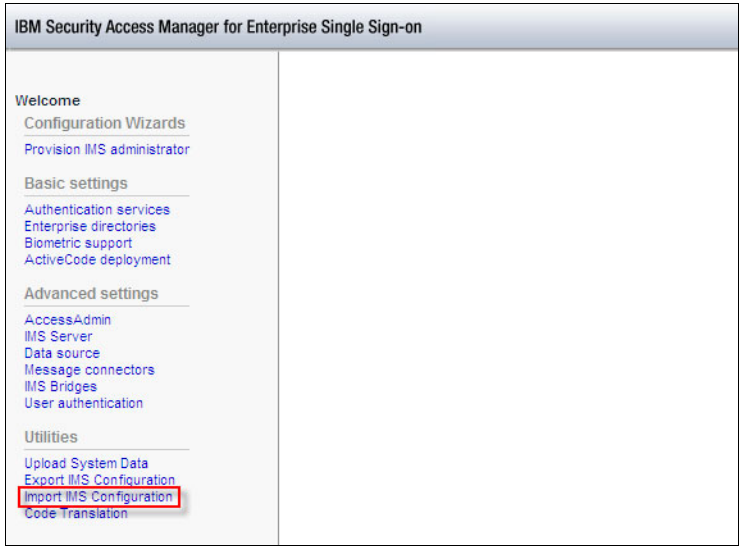

Figure 9-33 IMS Configuration utility

3. You are presented with the Import IMS Configuration wizard, as shown in Figure 9-34. You must click **Browse** to select the previously exported IMS Server configuration file.

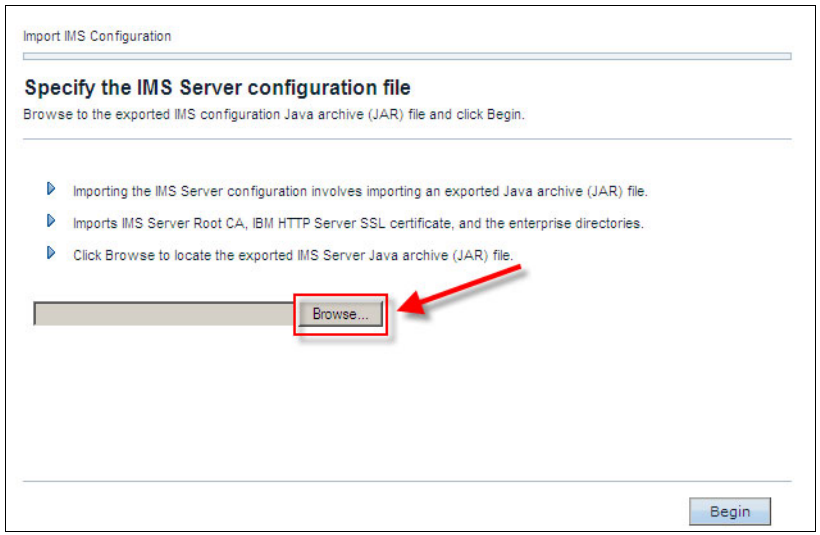

Figure 9-34 Import IMS Configuration wizard

4. Select the directory for the IMS exported JAR file, select the JAR file, and click **Open**, as shown in Figure 9-35.

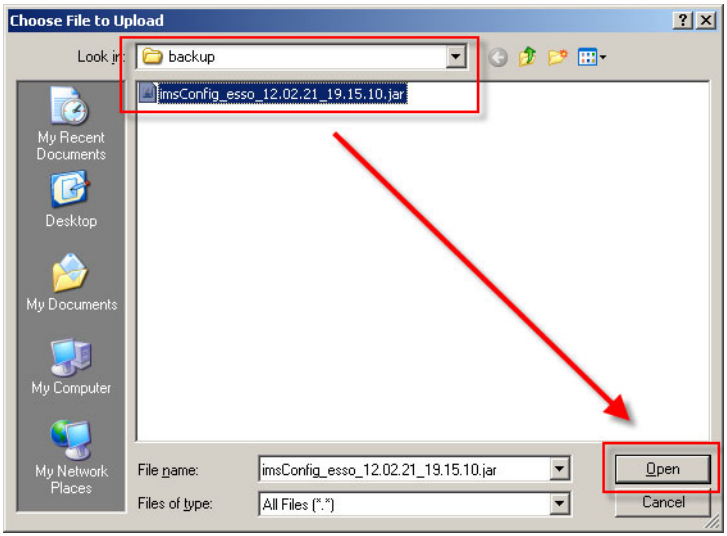

Figure 9-35 Select the exported IMS Server configuration JAR file

5. Click **Begin** to proceed with the import of the IMS Server configuration, as shown in Figure 9-36.

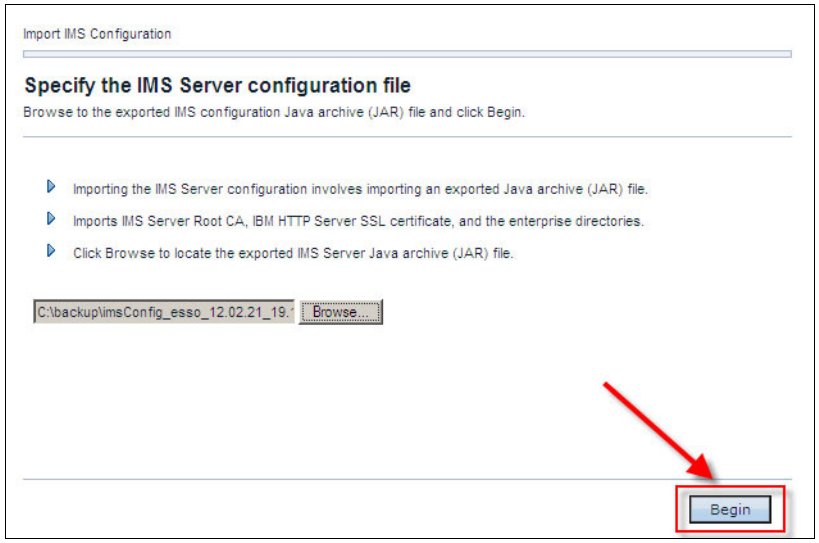

Figure 9-36 Import the IMS Server Configuration

6. Select the configuration to import, as shown in Figure 9-37, and click **Next**.

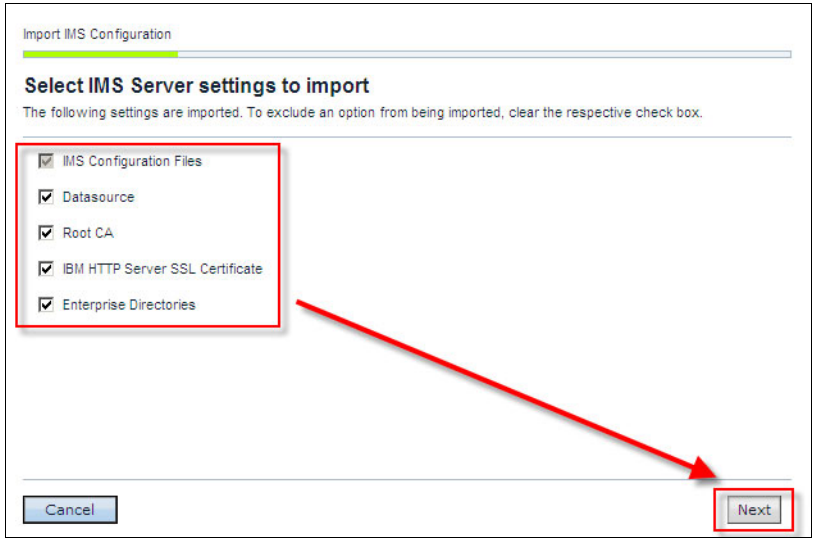

Figure 9-37 IMS Server settings to import

7. Accept the default client keystore and truststore details. The cardio healthcare company uses the default configuration of the IBM Security Access Manager for Enterprise Single Sign-On virtual appliance. Click **Next**, as shown in Figure 9-38.

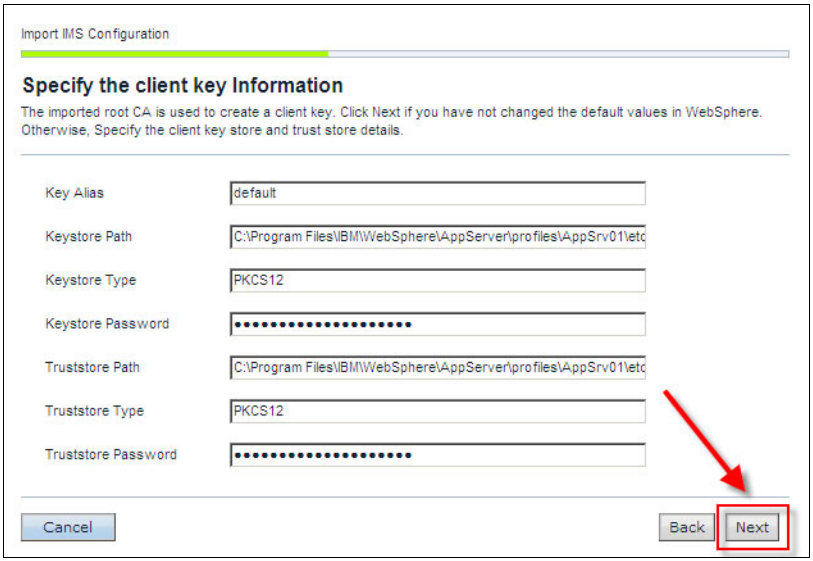

Figure 9-38 Client keystore and truststore

8. Accept the default IBM HTTP Server SSL Certificate details, and click **Next**, as shown in Figure 9-39 on page 297.

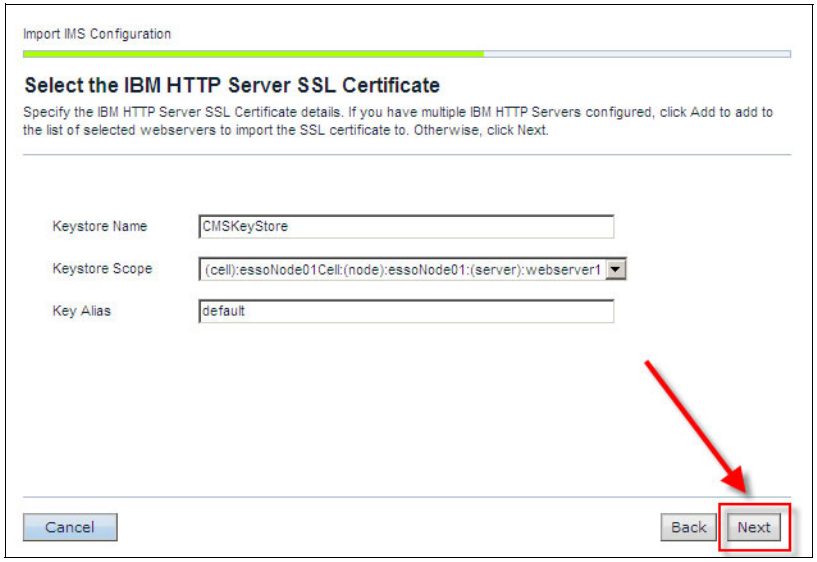

Figure 9-39 IBM HTTP Server SSL Certificate

9. Review the summary of the IMS Server configuration to be imported, as shown in Figure 9-40, and click **Import**.

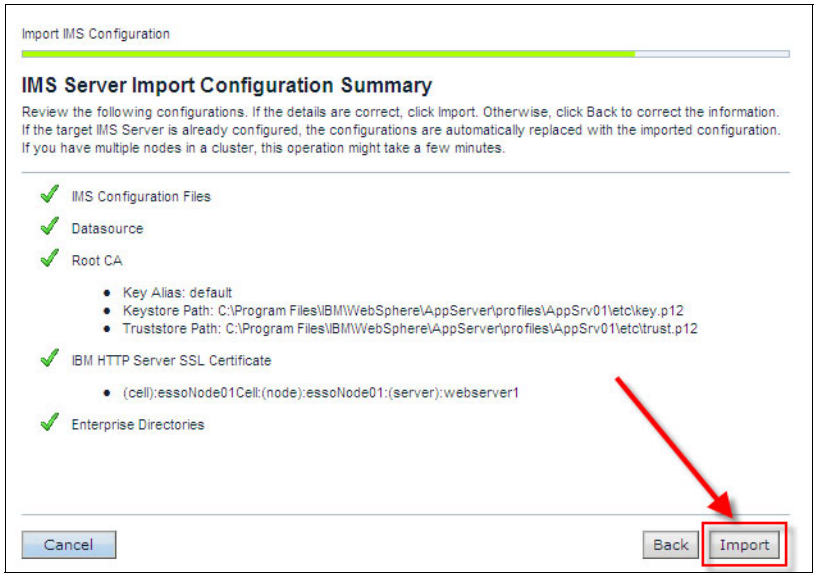

Figure 9-40 IMS Server Import Configuration Summary

10. Review the successful IMS Server Configuration report, as shown in Figure 9-41.

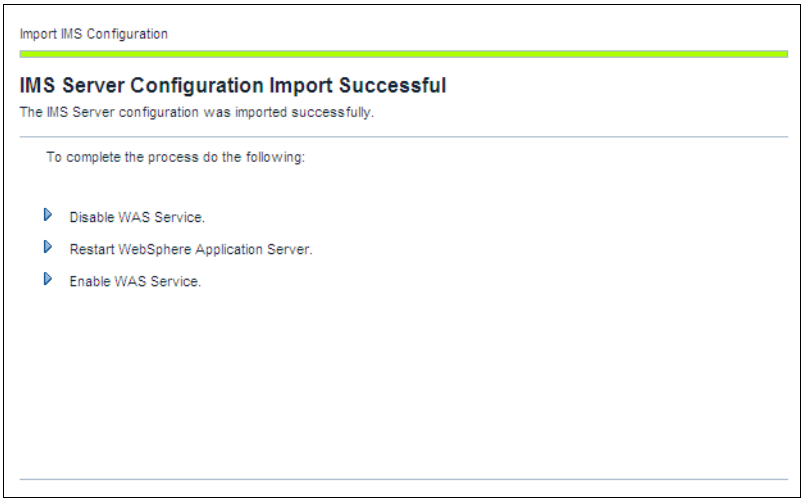

Figure 9-41 IMS Server Configuration Import Successful message

11. Stop the WebSphere Application Server by double-clicking the **Stop the application server (server1)** desktop icon, as shown in Figure 9-42.

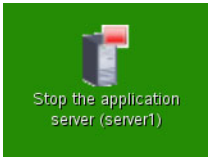

Figure 9-42 Stop the WebSphere Application Server

12. Stop the IBM HTTP Server by double-clicking the **Stop IBM HTTP Server** desktop icon, as shown in Figure 9-43.

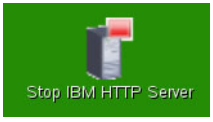

Figure 9-43 Stop the IBM HTTP Server

13. Start the IBM HTTP Server by double-clicking the **Start IBM HTTP Server** desktop icon, as shown in Figure 9-44 on page 299.

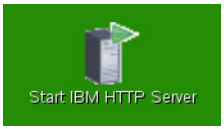

Figure 9-44 Start the IBM HTTP Server

14. Start the WebSphere Application Server by double-clicking the **Start the application server (server1)** desktop icon, as shown in Figure 9-45.

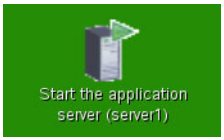

Figure 9-45 Start the WebSphere Application Server

Backup and restore information: For more information about backup and restore procedures, see the *IBM Security Access Manager for Enterprise Single Sign-On Version 8.2 Configuration Guide*, GC23-9692-01.

9.6 Tivoli Common Reporting

The Tivoli Common Reporting tool is a reporting feature available to users of Tivoli products, and it provides a consistent approach to view and administer reports. IBM Security Access Manager for Enterprise Single Sign-On provides auditing capabilities for its components. If auditing is enabled, the software generates audit events and stores them in the database. You can use Tivoli Common Reporting to produce reports about the audit events even if the IMS Server is not running. Tivoli Common Reporting generates reports in the HTML, PDF, Microsoft Excel, or Adobe PostScript format.

The cardio healthcare company uses the IBM Security Access Manager for Enterprise Single Sign-On virtual appliance. Tivoli Common Reporting components are already installed and activated in this virtual appliance. You can follow these steps to generate reports by using the Tivoli Common Reporting tool:

1. Log in to the virtual appliance with the default virtual image user ID that you created in step 12 on page 123. The cardio healthcare company created the user ID `virtuser`.

2. Start the Tivoli Common Reporting interface by double-clicking the **Tivoli Common Reporting Console** desktop icon, as shown in Figure 9-46.

Figure 9-46 Tivoli Common Reporting Console

3. The Mozilla Firefox browser opens the Tivoli Common Reporting Integrated Portal. In our example, as shown in Figure 9-47, the URL is `https://imsva.cardio.example.com:16311/ibm/console/logon.jsp`. Log on with the same user ID as in step 1 on page 299.

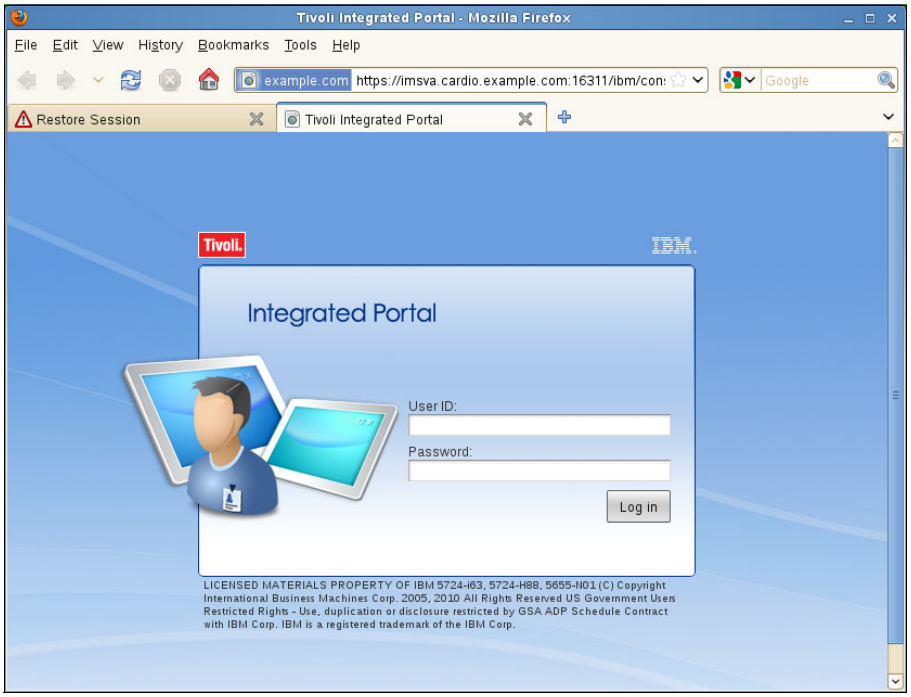

Figure 9-47 Tivoli Common Reporting logon screen

Note: You can also access the Integrated Portal from outside the IMS Server by directly opening the URL from step 3 on page 300.

4. If you open the Integrated Portal for the first time, you might see an `Untrusted Connection` warning. Expand the section **I Understand the Risks** and click **Add Exception**, as shown in Figure 9-48.

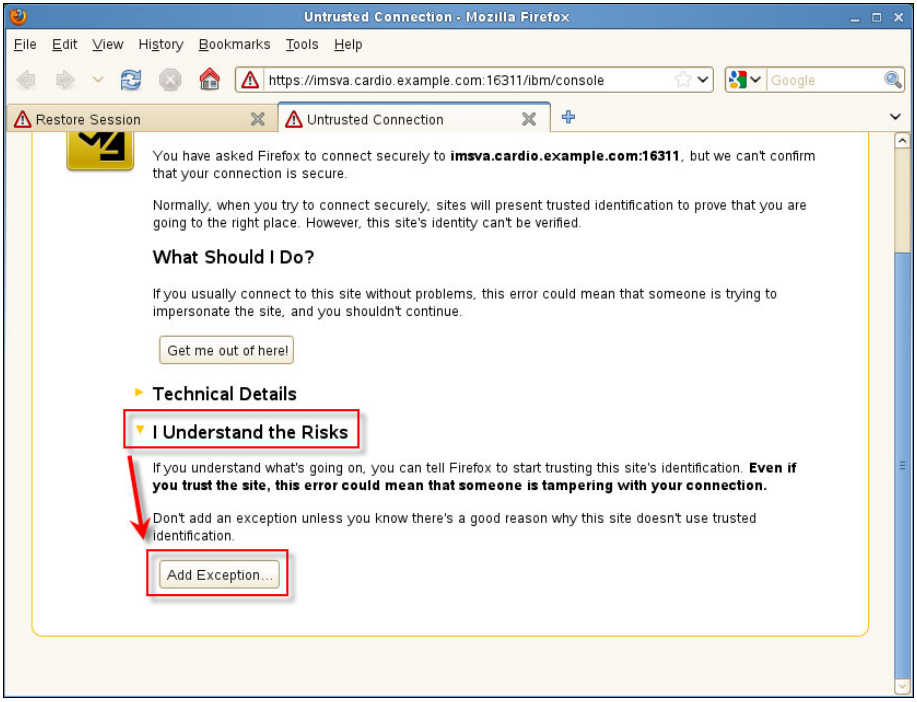

Figure 9-48 Untrusted Connection warning

5. Check **Permanently store this exception**, and click **Confirm Security Exception** (Figure 9-49 on page 302).

Figure 9-49 Add Security Exception

6. After you log on to the Integrated Portal, you see the Welcome window, as shown in Figure 9-50 on page 303.

   Note the product version:

Version	2.1.0.5
Build Number	cf171117.07, 201109271504
Build Date	4/28/11, 9/27/11

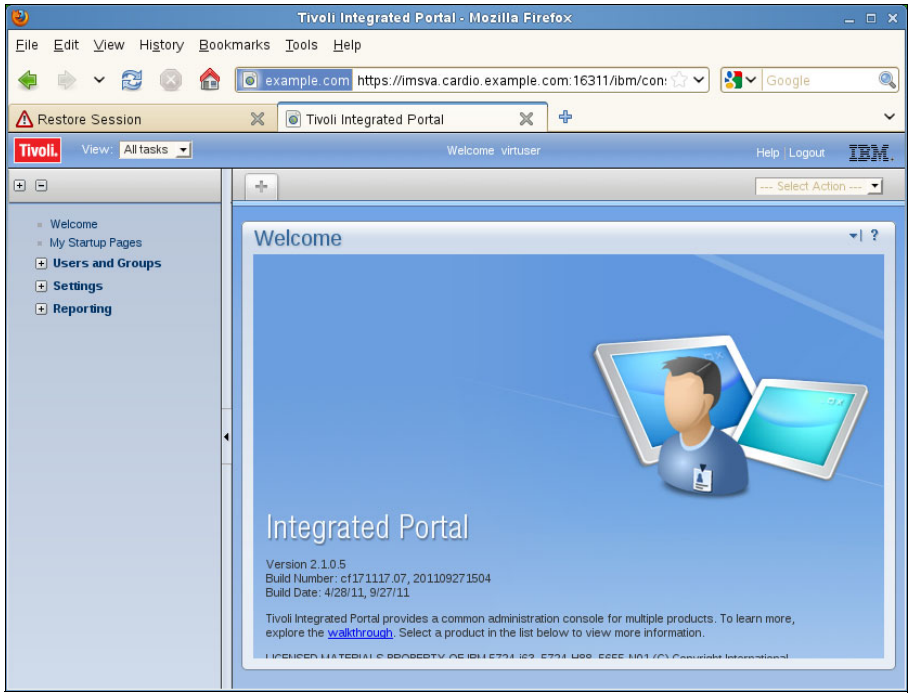

Figure 9-50 Integrated Portal Welcome window

7. Click **Reporting** → **Common Reporting** → **IBM Security Products**, as shown in Figure 9-51 on page 304.

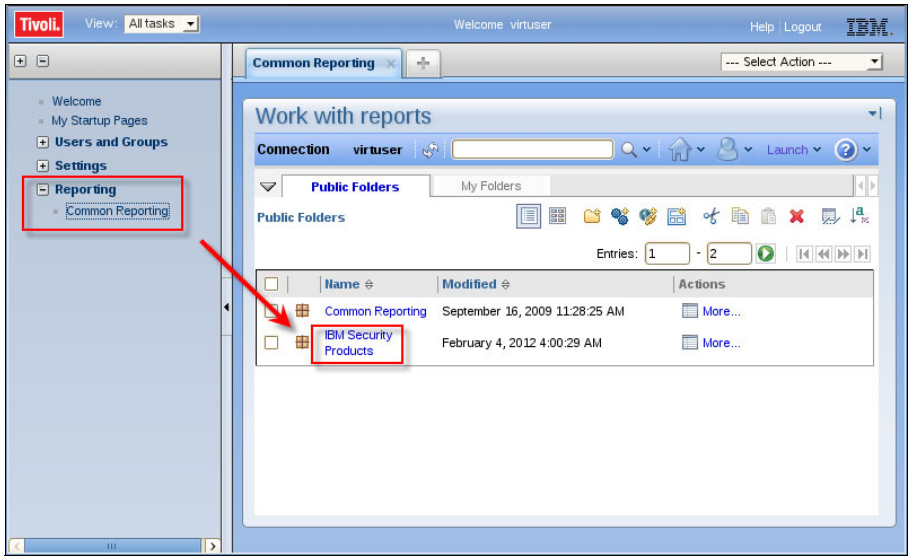

Figure 9-51 Open IBM Security Products reports

8. Click **SAM Enterprise Single Sign On v8.2** (Figure 9-52) to open the reports for IBM Security Access Manager for Enterprise Single Sign-On.

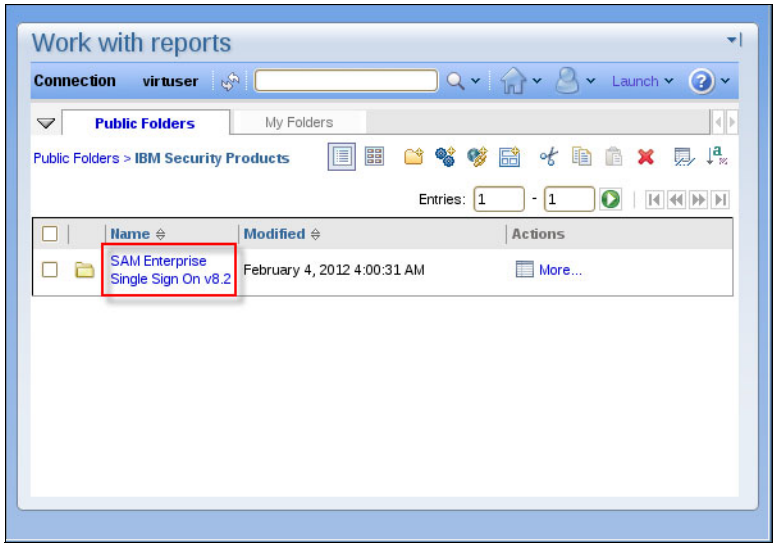

Figure 9-52 Open SAM Enterprise Single Sign On v8.2 reports

9. Next, you see an overview of the reports, as shown in Figure 9-53 on page 305.

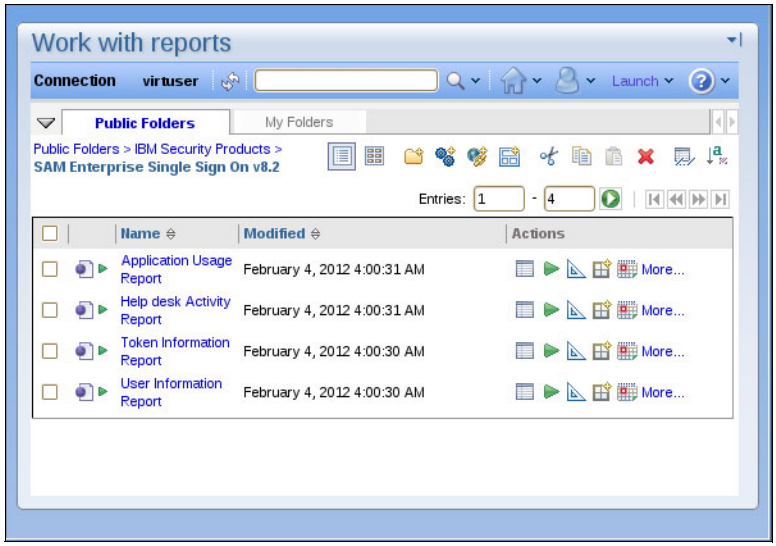

Figure 9-53 IBM Security Access Manager for Enterprise Single Sign-On reports

More information: For a description of all available reports, see Chapter 13. “Logging, auditing, and reporting” in the *IBM Security Access Manager for Enterprise Single Sign-On Planning and Deployment Guide Version 8.2,* SC23-9952-03.

For more information about Tivoli Common Reporting, see this website:

`http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tivoli.tcr.doc_211/ic-home.html`

10. Now, we run a sample report. Select **Application Usage Report** to open the input mask for the report, as shown in Figure 9-54 on page 306.

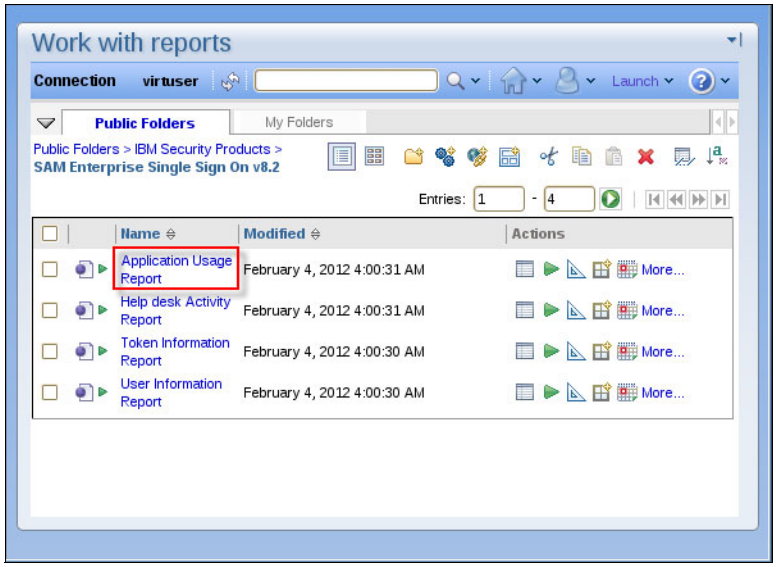

Figure 9-54 Open Application Usage Report

11. On the Work with reports window, type `notes` in the Authentication Service field. Keep ***** for all users in the User Name field. Then, select **Auto-capture authentication service password** to identify all captured passwords for the Lotus Notes AccessProfile, as shown in Figure 9-55.

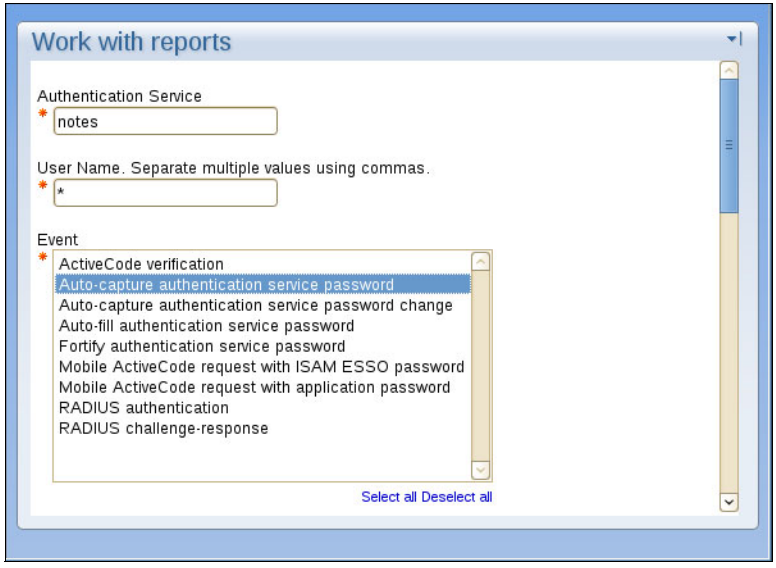

Figure 9-55 Configure report

12. Scroll down and adjust the date range (Figure 9-56).

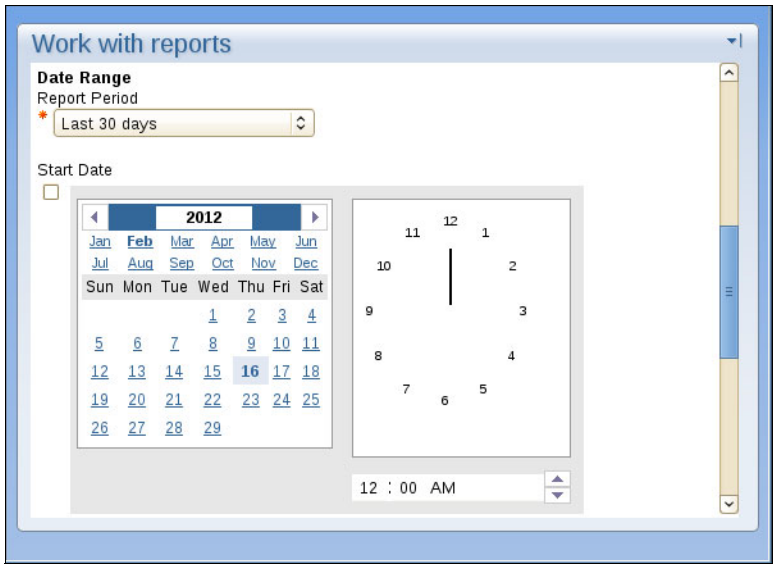

Figure 9-56 Configure report

13. Scroll down and click **Finish** to run the report, as shown in Figure 9-57.

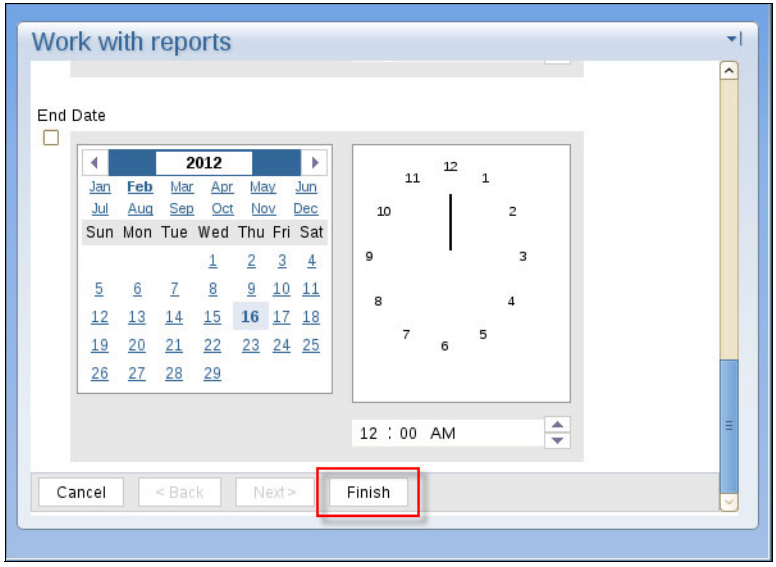

Figure 9-57 Finish report

14. Figure 9-58 shows the report results. The Notes password was captured five times for the Notes user `Administrator` and one time for the Notes user `Dr Bob`. We also see the authentication service, event, result, time of activity, and user machine IP address (origin of the event).

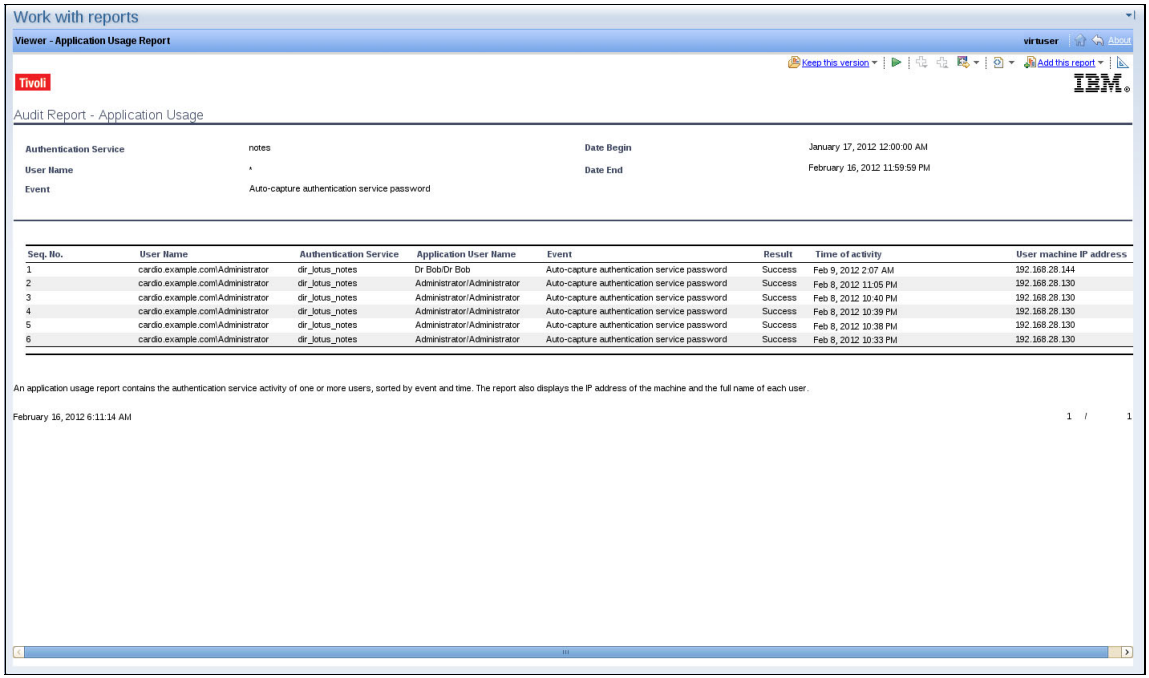

Figure 9-58 Sample report

9.7 Conclusion

In this chapter, we explained how the cardio healthcare company implements its operational requirements for IBM Security Access Manager for Enterprise Single Sign-On. We explained how to identify and deploy the available fixes. Then, we described the importance of maintaining the database audit logs to improve performance. We illustrated the disaster recovery procedures. We described how to generate operational reports by using the Tivoli Common Reporting tool.

Part 3

Appendixes

In this part, we provide information about advanced profiling and the configuration for strong authentication.

A

Renewing the Secure Sockets Layer certificate used by the IBM HTTP Server

This appendix demonstrates the necessary steps to renew a certificate if it is compromised or is about to expire.

The default expiration of a certificate is one year. Renewing a certificate re-creates all the information from the original certificate except the expiration date and key pair. The renewed certificate contains a new expiration date, and a public or private key pair. If the certificate for signing the chained certificate is not in the root keystore, you must use the default root certificate to renew the certificate.

Procedure to renew a certificate

To configure a new certificate, follow these steps:

1. Select **Start** → **All Programs** → **IBM WebSphere** → **Application Server** <***version***> → **Profiles** → <***profile name***> → **Administrative console**.
2. Log on to the IBM Integrated Solutions Console, as shown in Figure A-1.

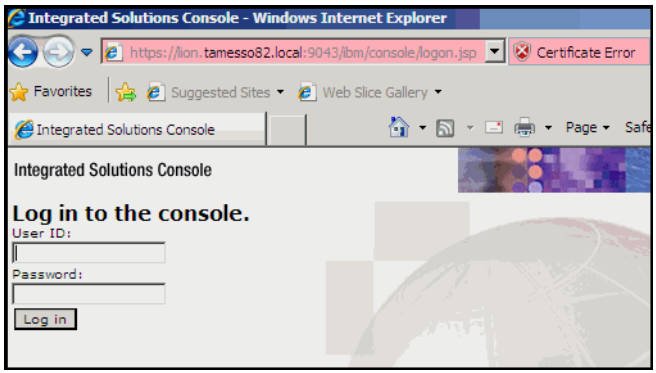

Figure A-1 IBM Integrated Solutions Console login window

3. On the IBM Integrated Solutions Console navigation pane, click **Servers** → **Server Types** → **Web servers**.
4. Click the <*web server name*> link, for example, **webserver1**, as shown in Figure A-2.

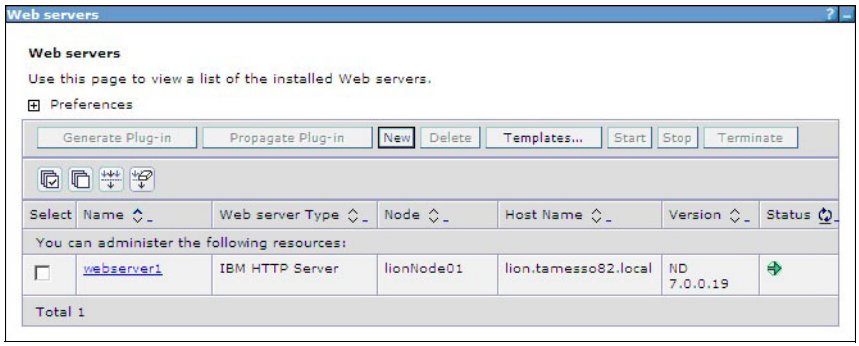

Figure A-2 Web server configuration

5. In the Additional Properties section on the Configuration tab, click **Plug-in properties**, as depicted in Figure A-3 on page 313.

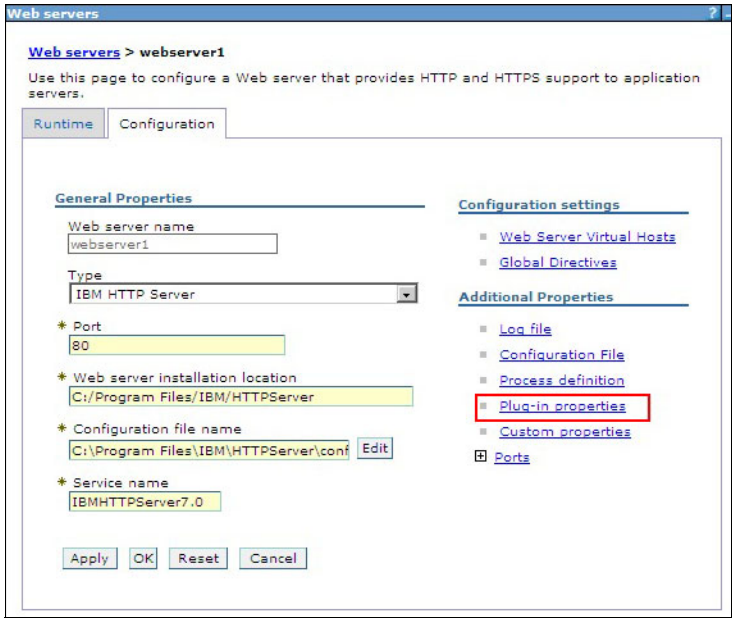

Figure A-3 Configuration tab for webserver1

6. In the Repository copy of the Web server plug-in files section, click **Manage keys and certificates**, as shown in Figure A-4 on page 314.

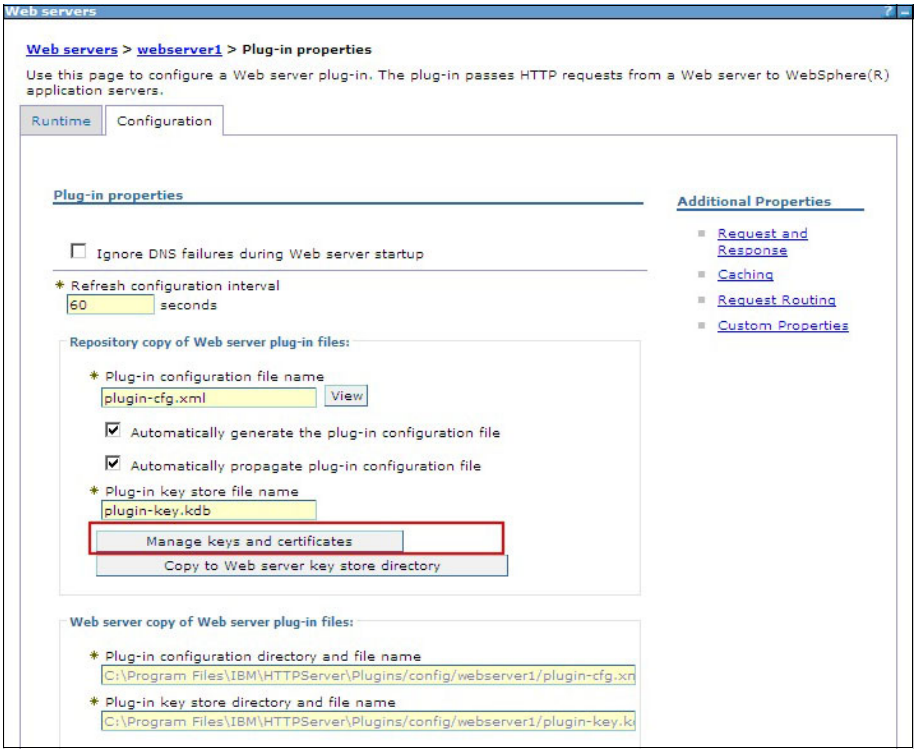

Figure A-4 Plug-in properties for webserver1

7. Under the Additional properties list, click **Personal certificates**.
8. In the following list, depicted in Figure A-5 on page 315, select the check box for the **default** Alias and click **Renew**.

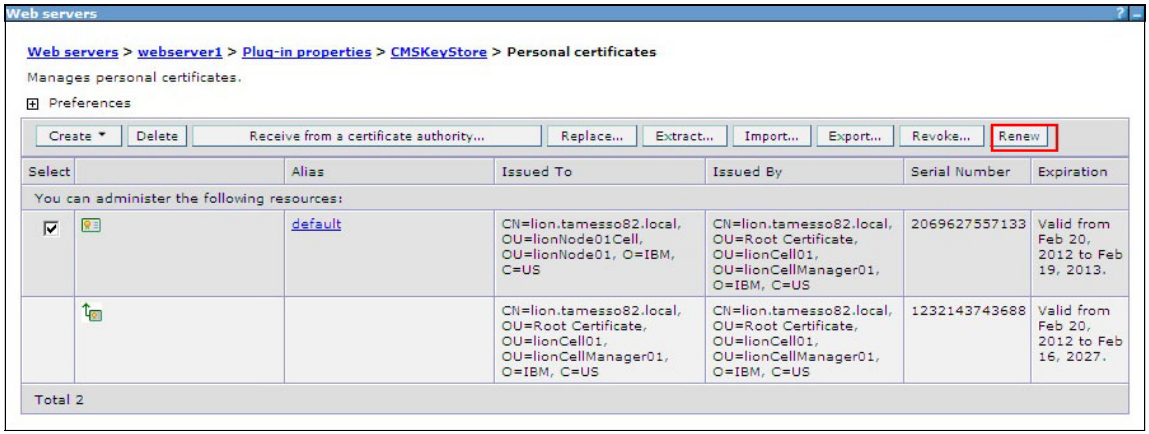

Figure A-5 Renew the default certificate of webserver1

9. Click **Save directly to the master configuration**, and click the **Plug-in properties** link.
10. Under the Repository copy of Web server plug-in files section, click **Copy to Web server key store directory**.
11. For WebSphere Application Server Network Deployment, you must follow the substeps listed; otherwise, skip to step 12 on page 316:
    a. Propagate the web server plug-in configuration:

> **Cluster environment:** In a cluster environment, you want all requests to come through one *central connection point*, so a single server URL is used. To define a central connection point, you must regenerate and propagate the WebSphere Application Server plug-in configuration for each web server.

    i. Click **Servers** → **Server Types** → **Web Servers**.
    ii. Select the listed web servers from the list, for example, **webserver1**.
    iii. Click **Propagate Plug-in**.
    b. Resynchronize the nodes with the deployment manager:
    i. On the IBM Integrated Solutions Console navigation pane, click **System administration** → **Nodes**.
    ii. Select all nodes that are managed by this WebSphere Application Server Network Deployment for the IMS Server, and click **Full Resynchronize**, as shown in Figure A-6 on page 316.

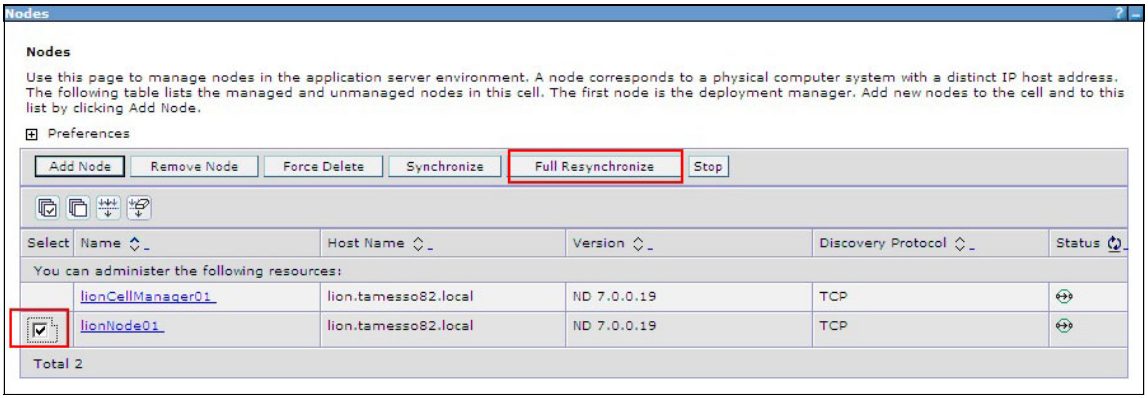

Figure A-6 Full Resynchronize on all the nodes

12. Restart the IBM HTTP Server:

    a. On the IBM Integrated Solutions Console navigation pane, click **Servers** → **Server Types** → **Web servers**.

    b. Select all the listed web servers, and click **Stop**.

    c. Click **Start** after all the web servers are stopped.

    d. In our example, we have a single server, `webserver1`, as shown in Figure A-7.

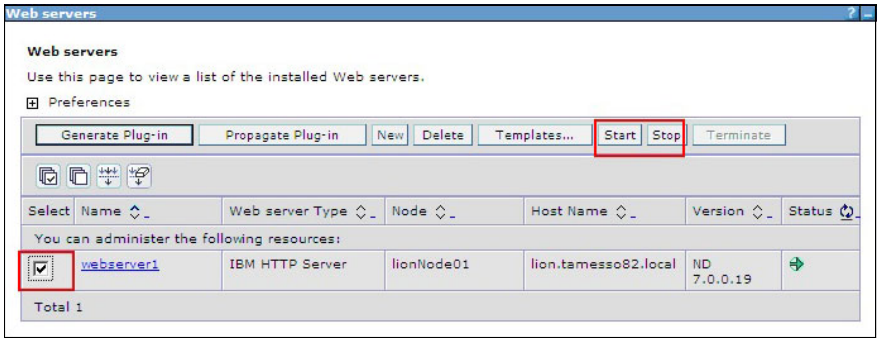

Figure A-7 Restart IBM HTTP Server

The necessary tasks are complete.

B

Advanced profiling

IBM Security Access Manager for Enterprise Single Sign-On enables users to access all their applications, including web, desktop, and heritage, and network resources, with the use of a single strong password. The solution helps simplify password management, protects information with strong authentication, and secures kiosks and shared workstations.

IBM Security Access Manager for Enterprise Single Sign-On helps strengthen security and meet regulations through stronger passwords and an open authentication device interface with a wide choice of strong authentication factors. It also facilitates compliance with privacy and security regulations by using centralized auditing and reporting capabilities.

In this appendix, we take a closer look at how to integrate web-based applications into IBM Security Access Manager for Enterprise Single Sign-On by using its AccessProfile technology. Toward the end of the appendix, we also investigate two non-web-based application profiles.

This appendix is a good resource for security administrators who are responsible for configuring and integrating IBM Security Access Manager for Enterprise Single Sign-On into the IT infrastructure of their organization.

This appendix contains the following sections:

- “Background” on page 319
- “Document complete event and the Observer” on page 322

- “Signatures” on page 325
- “Auto-learn AccessProfile” on page 329
- “Handling basic authentication” on page 329
- “Frames and the web browser document object” on page 330
- “Differences between Firefox and Internet Explorer AccessProfiles” on page 345
- “Common issues” on page 346
- “Use case” on page 359

Background

A typical modern web browser consists of multiple tabs, each of which can independently host a web page. A *web page* is a document written in HTML format and can be recognized by the starting `<html>` and ending `</html>` tags in the page source of the web page, typically available by the right-click context menu or file menu of the browser. It is possible for the browser to show other formats, such as XML, images, text, and PDF, but they are not relevant from a profiling point of view.

When writing AccessProfiles, you do not need to care about how the HTML page is generated on the server side by using Active Server Page (ASP), Common Gateway Interface (CGI), and JavaServer Pages (JSP). However, client-side JavaScript included in the web page can be important while writing an AccessProfile. Client-side JavaScript can modify the workflow of the web page within the browser while the AccessProfile acts on it.

HTML

HTML is a hierarchical format and can be thought of as a tree of HTML elements that serve as nodes.

The top-level node is HTML, which consists of a HEAD and a BODY node, as shown in Example B-1.

Example: B-1 Sample HTML

```
<html>
<head>
<title>Dummy page</title>
</head>
<body></body>
</html>
```

Example B-2 shows that each element can have optional attributes that qualify the way that the element is displayed. Here, the INPUT element has `ID` and `type` attributes.

Example: B-2 Sample HTML with attributes

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<title>Dummy page</title>
</head>
<body>
```

```
<input id="uname" type="text"></input>
</body>
</html>
```

HTML and JavaScript

Elements can also have events assigned to them that can affect the way that the HTML page behaves. JavaScript is the most common scripting language used to describe what must happen when the event occurs. Example B-3 shows an example.

Example: B-3 Sample HTML with JavaScript

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<title>Dummy page</title>
</head>
<body>
<input id="button1" type="button" onclick="javascript:alert("button
clicked");" />
</body>
</html>
```

In Example B-3, the BUTTON element has an `onclick` event, which when the button is clicked, shows a button clicked message by using JavaScript.

For non-trivial cases, the event is invoked as a function call, such as the function call shown in Example B-4.

Example: B-4 Sample HTML with JavaScript function call

```
<html>
<head>
<title>Untitled Page</title>
</head>
<body>
<script type="text/javascript">
function clicked() {
alert("button clicked");
}
</script>
<input id="button1" type="button" onclick="clicked();" value="Click me"
/>
</body>
```

```
</html>
```

JavaScript and DHTML

It is possible for an HTML element event's JavaScript to modify other elements of the page, making the HTML more dynamic, therefore, the name *Dynamic HTML* (DHTML). It is important to watch for cases, such as these cases, because they can lead to unexpected single sign-on (SSO) behavior. See Example B-5.

Example: B-5 Sample HTML with JavaScript

```
<html>
<head>
<title>Untitled Page</title>
<script type="text/javascript">
function body_loaded() {
uname.value = "";
}
</script>
</head>
<body onload="body_loaded();">
<input id="uname" type="text" value="start-val" />
</body>
</html>
```

In this case, if the AccessProfile injected anything in the `uname input` field before the onload event on BODY element was called, the text is cleared by the **`uname.value=""`** command, which gives the impression that no injection occurred.

HTML load sequence

When a user types a URL in the address bar and submits it, the following events occur:

- The web page is downloaded. While the web page is being downloaded, the browser starts to parse the html page and starts to partially show its content on the screen. The Observer ignores this downloading process altogether, because the browser does not provide any useful events at this time.
- The web page completes downloading and the browser finishes parsing through the HTML content and creates an internal Document Object Model (DOM) object that is a binary representation of the HTML elements, their attributes, and scripts. Only after this DOM object is created does the browser

inform the Observer about the existence of this page in the form of a document complete (the DOM object is now fully formed and available) event.

This event is trapped by the Observer and available as a web page completes loading trigger and used internally to determine whether it needs to load a new AccessProfile for the web page.The details of the mechanism are explained in “Document complete event and the Observer” on page 322.

- The onload event for the BODY element fires and executes any onload script that is specified in the HTML page. This event is trapped by the Observer and made available as the HTML element completes loading the trigger. This trigger is a generic trigger and can also be used for tracking HTML element completes that load events in other elements, such as FRAME and IMG.

Document complete event and the Observer

The document complete trigger plays an important role in determining the lifetime and behavior of an AccessProfile instance within the browser.

The first document complete event

We start from the time that the browser starts a new tab and submits a URL. At the first document-complete, the web-sso-agent notices that there is no profile loaded currently and it contacts the DataProvider and passes to it the details of the current web page, such as domain, path, protocol, and port.

The DataProvider matches the passed information against the signatures of the currently loaded AccessProfiles. Four things can happen at this stage:

- The DataProvider finds multiple AccessProfiles for the passed information. To avoid unpredictable behavior, the Observer does not load any of the matched AccessProfiles, but instead writes an error line in the Observer logs and in the General tab of AccessStudio messages.
- The DataProvider finds a single match for the passed information. The matched AccessProfile ID is returned to the web-sso-agent, which then calls to fetch the AccessProfile object and creates the state-machine object described in it.

  In AccessStudio, if there is not an existing tab open for the browser, a tab is created and a new log with the following log name is shown:

  `Loaded AccessProfile:` *profile-name*
- The DataProvider does not find any matching AccessProfiles, but there is an auto-learn AccessProfile defined. In this case, the auto-learn AccessProfile ID

is returned to the web-sso-agent as a match, which then fetches the actual AccessProfile object and creates the state-machine object described in it.

In AccessStudio, if there is not an existing tab open for the browser, a tab is created and a new log with the following log is shown:

```
Loaded AccessProfile: sso_site_web_auto_learn
```

- The DataProvider does not find any matching AccessProfile and there is no auto-learn AccessProfile defined (or the auto-learn is disabled by a policy). Considering that the Observer passes this first document-complete event to the newly loaded state-machine as well.

Based on the document-complete triggers that are available in the start-state, either one of those triggers can fire, moving the state-machine to the next state or it can stay in the same place if no trigger fires. The exception is the auto-learn AccessProfile. If the starting web page completes loading trigger does not fire, the state-machine is unloaded immediately. The flow diagram in Figure B-1 on page 324 explains this behavior.

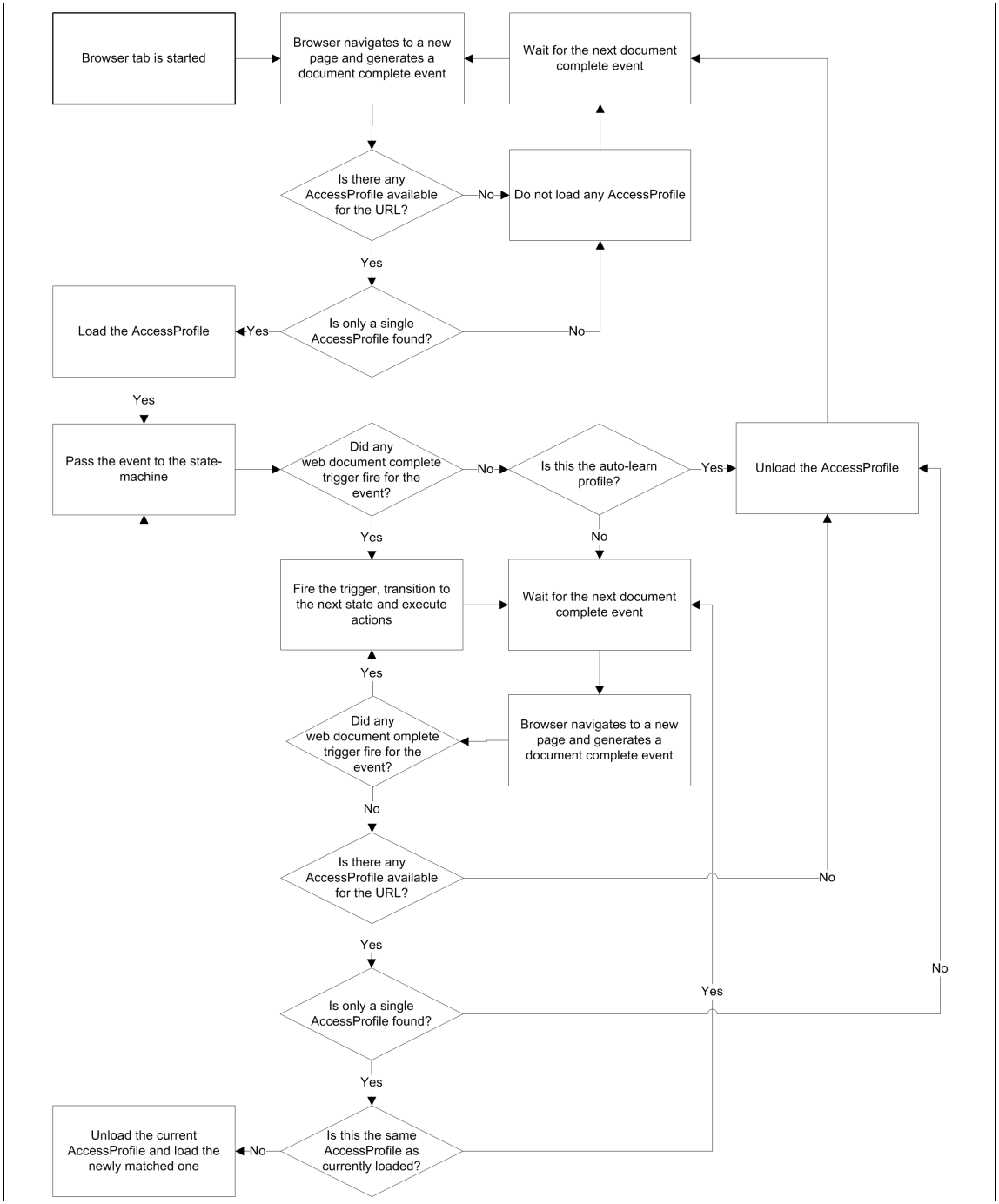

Figure B-1 State machine behavior w.r.t. document complete events

In AccesStudio, there is no log for this event, so if there is no existing tab for the browser process, there is no tab created at this time.

In addition to being used to load the AccessProfile, the document complete event is of interest to the AccessProfile writer because it is a good place to inject credentials.

Subsequent document complete events

After the first document complete was rerouted to the state-machine after it caused the loading of the AccessProfile, the rest of the document complete events are treated in the following fashion:

- If there is a webpage completes loading trigger in the current state that matches this document-complete-event, that trigger fires and the state-machine continues as normal.
- If there is no webpage completes loading trigger in the current state that matches this document-complete-event, the agent goes back to the DataProvider and fetches a matching profile. In this case, the fetched profile is the same as the current profile, and the state-machine stays where it is. Otherwise, if the fetched profile differs from the current profile, the current profile is thrown away, the new one is loaded, and the subsequent sequence is identical to the first document complete event previously described.

Signatures

The Observer uses two kinds of signatures for web applications: the web-signature and the HTML signature.

The web-signature

The web-signature, shown in Example B-6, is the site-signature for web applications. It is used to match the correct AccessProfile to a certain web application. It can be one level deep only, and it has the following form:

`/child::web[<attribute filters list>]`

Example: B-6 Sample web signature

```
https://www.google.com/accounts/ServiceLogin?service=mail&passive=true&
rm=false&continue=http%3A%2F%2Fmail.google.com%2Fmail%2F%3Fui%3Dhtml%26
zy%3Dl&bsv=llya694le36z&scc=1&ltmpl=default&ltmplcache=2
```

By using the example shown in Example B-6 on page 325, the list of attributes in Example B-7 through Example B-12 on page 327 are valid. A string attribute requires quotation marks around it when specified as a filter in the signature. A numeric attribute does not have quotation marks. These attributes are valid:

- **domain** (string): This attribute is the text between the '//' after the protocol and the first '/'.

  Example: B-7 Sample domain

  ```
  /child::web[@domain="www.google.com"]
  ```

- **protocol** (string): This attribute can be http, https, file, or ftp.

  Example: B-8 Sample protocol

  ```
  /child::web[@protocol="http" and @domain="www.google.com"]
  ```

- **url** (string): The complete URL of the page, which is seen on the address bar of the browser.

  Example: B-9 Sample url

  ```
  /child::web[@url="https://www.google.com/accounts/ServiceLogin?servi
  ce=mail&passive=true&rm=false&continue=http%3A%2F%2Fmail.google.com%
  2Fmail%2F%3Fui%3Dhtml%26zy%3Dl&bsv=llya694le36z&scc=1&ltmpl=default&
  ltmplcache=2"]
  ```

  The url attribute is not used often, because the constituent parts of the url are available as other attributes, such as domain and query_string.

- **path** (string): This attribute is the text after the domain but before the querystring (the text that begins with a question mark (?)). It always starts with a forward slash (/).

  Example: B-10 Sample path

  ```
  /child::web[@domain="www.google.com" and @protocol="https" and
  @path="/accounts/ServiceLogin"]
  ```

  The path attribute is typically used in enterprise portal scenarios where paths hold several applications that require their own logins.

- **port** (numeric): If not specified, the value is `80` for web pages with the http protocol and `443` for the https protocol.

Example: B-11 Sample port

```
/child::web[@domain="www.google.com" and @protocol="https" and
@path="/accounts/ServiceLogin" and @port=443]
```

- **query_string** (string): This attribute is the text after and including the question mark (?) in the url.

  Example: B-12 Sample query_string

  ```
  /child::web[@domain="www.google.com" and @protocol="https" and
  @path="/accounts/ServiceLogin" and @port=443 and
  @query_string="?service=mail&passive=true&rm=false&continue=http%3A%
  2F%2Fmail.google.com%2Fmail%2F%3Fui%3Dhtml%26zy%3Dl&bsv=llya694le36z
  &scc=1&ltmpl=default&ltmplcache=2"]
  ```

  The query_string is almost always used with a regular expression check (~ or #) rather than directly, because it is too specific to be used on its own.

  Use it as a last resort for identifying a page, because it is a set of parameters passed to the page, and it can change from one machine (and browser) to the other.

The HTML signature

The *HTML signature* identifies the HTML elements. These elements are used as a part of a *web control* in AccessStudio to identify the HTML element to inject to or capture data from. These elements are used for identifying HTML elements for triggers, such as the HTML element clicked or HTML element is found.

The root '/' element is the HTML tag, so `/child::html` is at the level of the BODY and HEAD elements. The elements are identified by their type by using the attribute `tag_name`:

```
/child::html[@tag_name="body"]
```

The value of the `tag_name` attribute is case-insensitive, so both of the examples in Example B-13 work.

Example: B-13 Sample tag_names

```
/child::html[@tag_name="body"]
/child::html[@tag_name="BODY"]
```

Unlike window signatures that have a predefined list of properties that can be used as attributes, each HTML element can have its own set of properties. For example, an INPUT element has a `type` property, which describes whether it is a

text-box or a button. An “A” (anchor) element has an `href` property, which defines the URL to which the user is directed when the link is clicked.

These properties can be used as attributes while defining a signature (Example B-14).

Example: B-14 Sample HTML properties

```
/descendent::html[@tag_name="input"and @type="button" and @id="submit"]
```

HTML elements can have events described for them too, for example, an INPUT element can have an *onclick* method. These methods cannot be used as attributes while writing a signature.

Beyond mere identification of HTML elements, HTML signatures can also be used to retrieve data from a certain HTML element. By default, a web control (such as the web controls used in injection, capture, or data transfer actions) extracts the `value` data of the specified HTML element. This extraction can be explicitly overridden to get other values, such as the inner text, inner html, or any other defined property for that control (Example B-15).

Example: B-15 Sample HTML properties

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<a href="http://localhost/d/a.html">Site A</a>
</body>
</html>
```

For example, the following signature returns 'Site A'. This signature can be useful to extract text from the web page, which can be used for auditing purposes or other workflow decisions within the AccessProfile:

```
/descendent::html[@tag_name="a" and
@href#"http://localhost/d/a\.html.*"]/@inner_text
```

Like `inner_text`, `inner_html` can be used to return the HTML embedded within the identified HTML element; however, it has less real-world utility.

In fact, any attribute that is defined for an HTML element can be retrieved in this way. For example, the following example for the HTML sample in Example B-15 returns '`http://localhost/d/a.html`':

```
/descendent::html[@tag_name="a" and
@href#"http://localhost/d/a\.html.*"]/@href
```

Auto-learn AccessProfile

IBM Security Access Manager for Enterprise Single Sign-On comes bundled with a default profile for web applications, which is loaded if there is no explicitly defined AccessProfile found for a web page. This AccessProfile relies on the fact that most website login pages follow a standard pattern of a user name and a password input element that is embedded in a form element.

The auto-learn AccessProfile is identified by a special signature, such as this signature:

```
/child::web[@domain="*"] (note the ="*" for domain)
```

It also contains a generic set of user name and password signatures, such as these user name and password signatures:

```
/child::html/child::html/descendent::form/descendent::input[@type="pass
word"]/ancestor::form/descendent::input[(@tag_name="input") and
(@type="text" or @type="")]
```

We provide another example:

```
/child::html/child::html/descendent::form/descendent::input[@tag_name="
input" and @type="password"]
```

Certain clients might not want to turn on this auto-learn behavior. It can be turned off by following these steps:

1. Setting IMS scope policy – pid_sso_auto_learn_enabled to `false` (suggested)
2. Removing sso_site_web_auto_learn AccessProfile from IMS

Handling basic authentication

Basic authentication is a mechanism where the authentication to the web server is done as a part of the exchange of HTTP header information rather than through an HTML login form.

The authentication happens before the web page contents are downloaded. Therefore, the common approach of automating the HTML login page does not work because there is no HTML with which to work.

Instead, the browser shows its own windows dialog, which asks for the credentials for the website. The server to which the browser connects is shown as a part of the windows dialog UI.

There is no need to write an AccessProfile to handle basic authentication. The standard AccessProfile shipped with IBM Security Access Manager for Enterprise Single Sign-On has the relevant workflow to extract the server locator from the basic authentication dialog.

For Internet Explorer 8.0, the server locator is extracted by the regular expression `– Connect to (.*)`. You must associate this server locator to the right authentication service. For other browsers, use different regular expressions to extract the server locator.

Frames and the web browser document object

In web browser terminology, a *document* refers to an object that is responsible for displaying the contents of a web page within a browser tab. Each browser tab has its own unique document object. As the user navigates from one web page to another (either by clicking a link or by submitting the URL in the address bar), the same document object continues to download and display the web page content.

Only one AccessProfile can be loaded for a document at a certain time. However, as the document moves from displaying one web page to another, it can unload the existing AccessProfile and load new AccessProfiles. For example, as the user navigates from Gmail to Yahoo! Mail, the document unloads the Gmail AccessProfile (if present) and loads the Yahoo! Mail AccessProfile (if present).

The reason why the document object matters as far as writing AccessProfiles is concerned is because it is possible for one document object to contain other document objects. And because AccessProfiles are loaded in the context of a document object, the AccessProfile writer can see what looks like multiple AccessProfiles loaded at the same time for a browser tab.

This multiple documents per browser tab scenario is what occurs if a web page uses frames or iframes. A frame (irrespective of whether it is defined in a FRAMESET or an IFRAME tag) is an instruction by the web page to the document object that displays it to create another child document object and load the contents of another web page within and show it at a particular place within the page layout. The web page of this child document object might in turn instruct it to create further child document objects to create a hierarchy of document objects that are each associated with a web page.

Example B-16 shows an example of two iframes that are loading.

Example: B-16 a.html (defines two iframes that load b.html and c.html)

```
<html>
<head>
<title></title>
</head>
<body>
This is is the main web page - A
<br />
<br />
<iframe src="b.html" height="100" width="300"></iframe>
<br />
<br />
<iframe src="c.html" height="100" width="300"></iframe>
</body>
</html>
```

Example B-17 shows an example of a login page where clicking the login button takes the document to another web page.

Example: B-17 b.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<script type="text/javascript">
function clicked() {
document.url="d.html";
}
</script>
This is is the contained web page - B
<br /><br />
User: <input type="text" id="username" />
<br /><br />
Pwd: <input type="text" id="pwd" />
<br /><br />
<input type="button" id="submit" onclick="clicked();" value="Login"/>
</body>
</html>
```

Example B-18 shows that a document (a.html) can contain more than one child document.

Example: B-18 c.html

```
<html>
<head>
<title></title>
</head>
<body>
This is is the contained web page - C
</body>
</html>
```

Example B-19 simulates the successful login screen.

Example: B-19 d.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
This is is the contained web page - D
<br /><br />
Successfully Logged in!
<br /><br />
<a href="b.html">Back to b.html (the login page)</a>
</body>
</html>
```

Figure B-2 on page 333 shows what an html looks like when loaded in the browser.

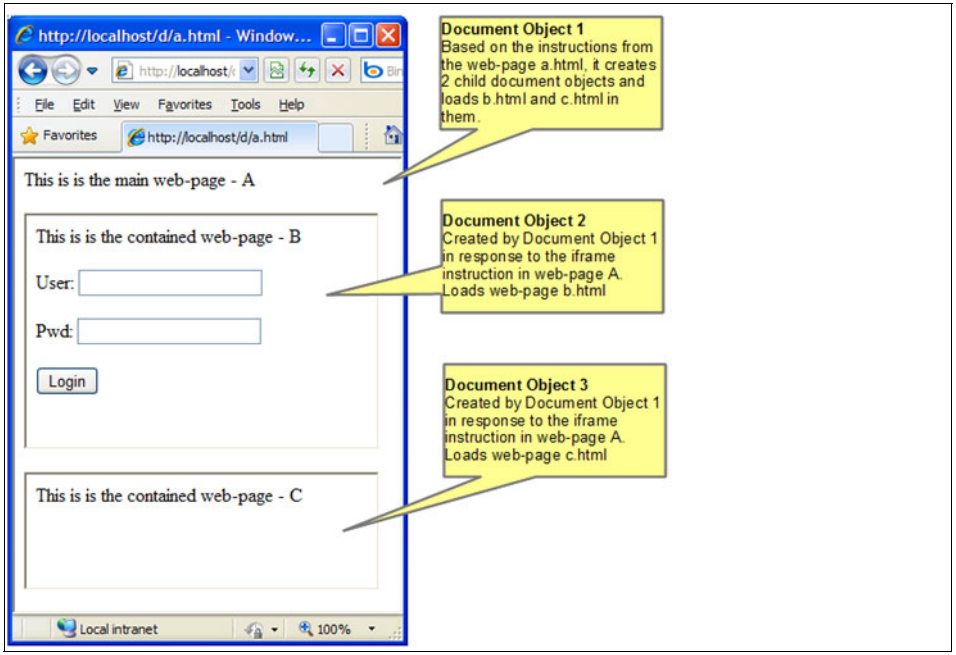

Figure B-2 Page a.html with frames

Consider the cases shown in the following sections.

Web page B navigates to another web page within Document 2

After navigation, the browser looks like Figure B-3 on page 334.

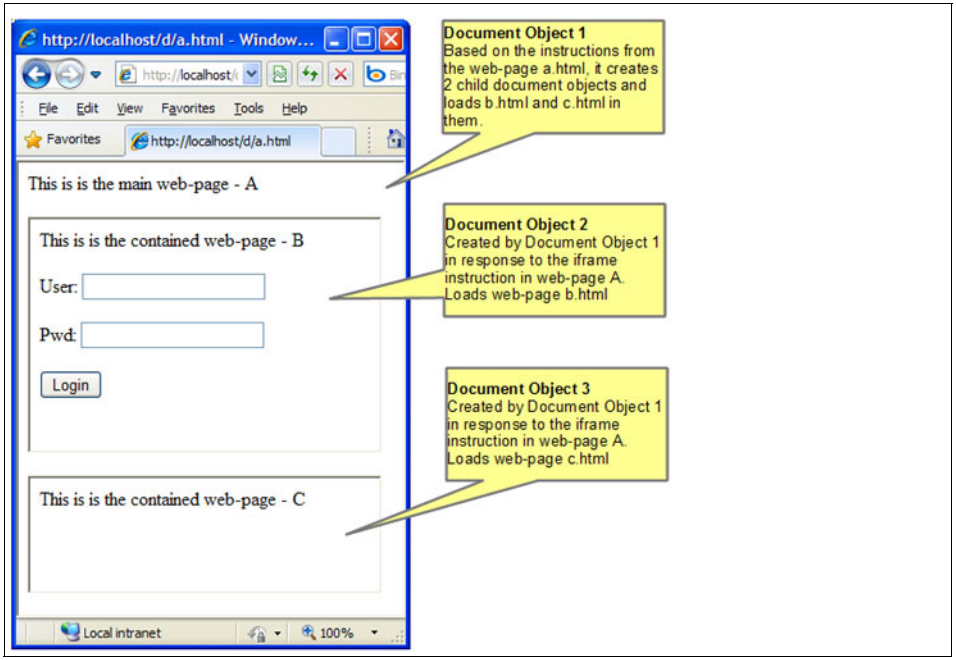

Figure B-3 Frame moved to c.html

From the writing an AccessProfile perspective, the fact that web pages B and D are in a frame and not in the main document object is irrelevant, because the document that hosts the two web pages is the same. The AccessProfile looks exactly like web page b.html was loaded directly in the browser and not under a frame, because the AccessProfile does not consider any document other than the document in which it resides.

In summary, as long as all of the workflow, in which the AccessProfile is interested, occurs in one document object, the AccessProfile writer can ignore whether the web pages in the workflow were hosted in a child document object or the main document object.

A child document web page causes navigation in the parent document

Consider the same starting document in the earlier case. Figure B-4 on page 335 shows page a.html with frames.

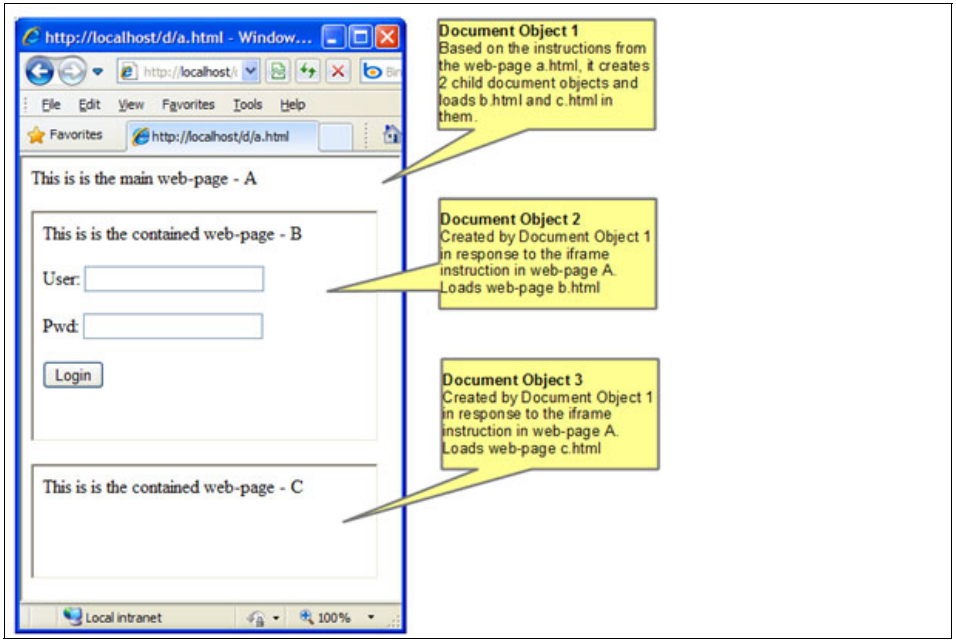

Figure B-4 Page a.html with frames

This time, when the user clicks **Login**, instead of document object 2 loading another web page, the main document 1 moves to another web page.

Example B-20 shows the modified `b.html`. Compare it to the `b.html` of case 1. The parent document (document 1) navigates this time and not the current document.

Example: B-20 Modified b.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<script type="text/javascript">
function clicked() {
window.parent.document.URL="e.html";
}
</script>
This is is the contained web page - B
<br /><br />
User: <input type="text" id="username" />
<br /><br />
```

```
Pwd: <input type="text" id="pwd" />
<br /><br />
<input type="button" id="submit" onclick="clicked();" value="Login"/>
</body>
</html>
```

Example B-21 shows the markup for `e.html`.

Example: B-21 e.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
This is is the web page E
<br /><br />
Successfully Logged in!
<br /><br />
<a href="a.html">Back to a.html</a>
</body>
</html>
```

Figure B-5 shows what displays when the user clicks **Login**.

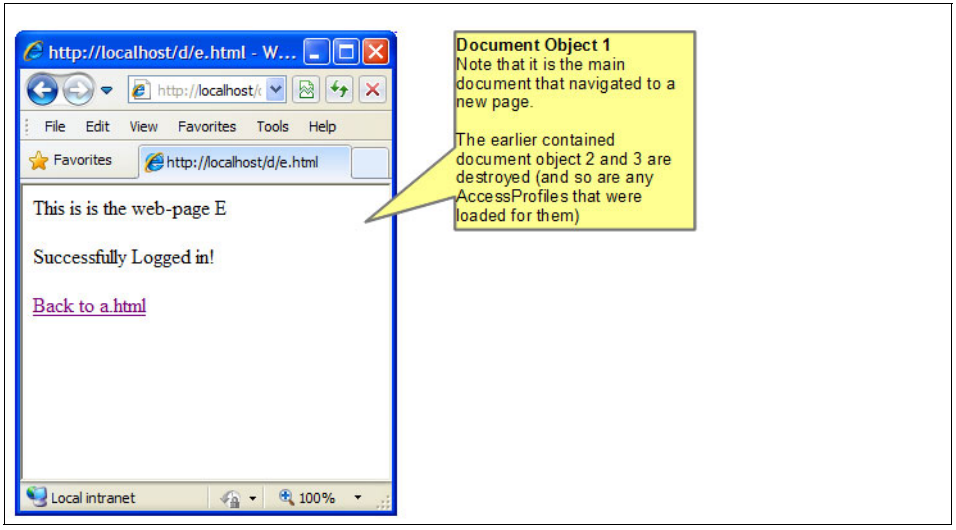

Figure B-5 The main document navigated to a new page

If there was a non-validating AccessProfile for `b.html` (the capture and save of credentials happened on the Login button click event), the workflow, in which the AccessProfile is interested, is complete before document object 2 (and the attached SSO AccessProfile) is destroyed.

However, an AccessProfile, which was trying to validate a login and was thus waiting for `e.html` to confirm that the login happened successfully, cannot work because it gets destroyed on navigation of the main document object, which destroys all child document objects (including the one to which the AccessProfile was attached).

The typical symptom of this problem in an actual environment is that the logs show a successful firing of the html click trigger and the capture action, but the following web page completes loading the trigger with the save action that never happens. Instead, the logs might show AccessProfile loading logs (typically of the same AccessProfile or of auto-learn), because the main document object, when navigating to the new page, might be loading new frames for the successful login page, which requires their own AccessProfile.

Solution 1

One work-around is to use a global account-data-bag (set Use local bag = No) to temporarily store the captured credential, so it can be retrieved later by another AccessProfile, which is meant to detect the successful login page, as shown in Figure B-6.

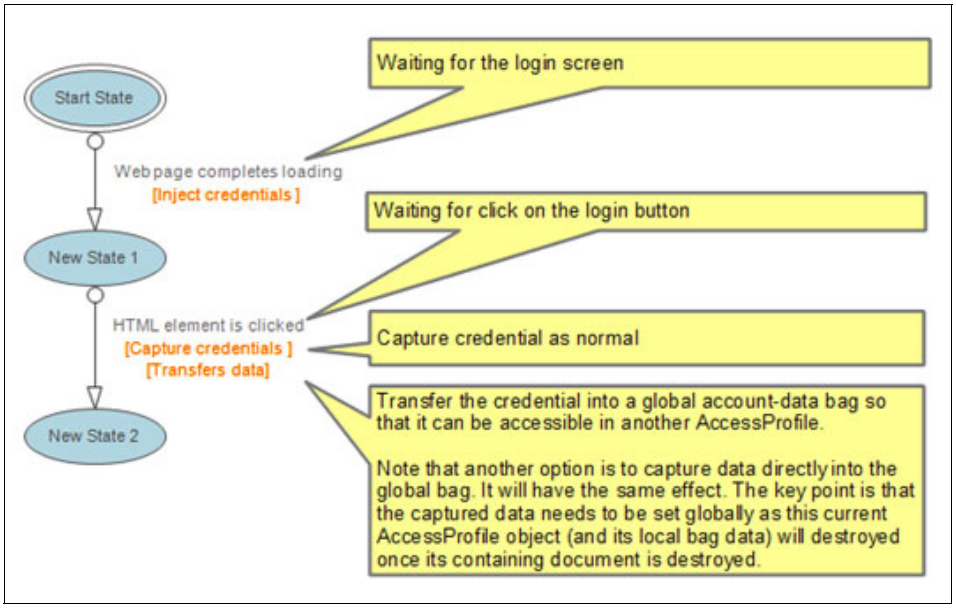

Figure B-6 AccessProfile for capturing credential

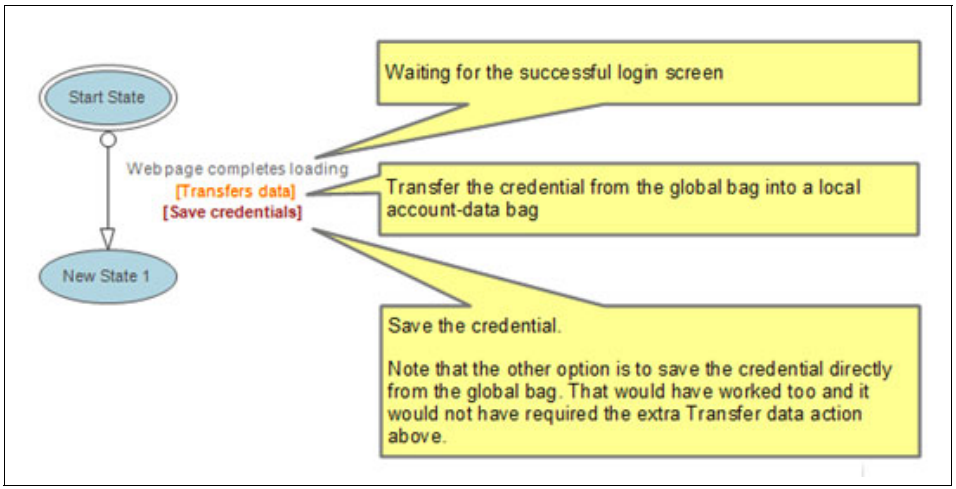

Figure B-7 AccessProfile for saving credential

As an optimization, the two AccessProfiles can be combined into one, as shown in Figure B-8.

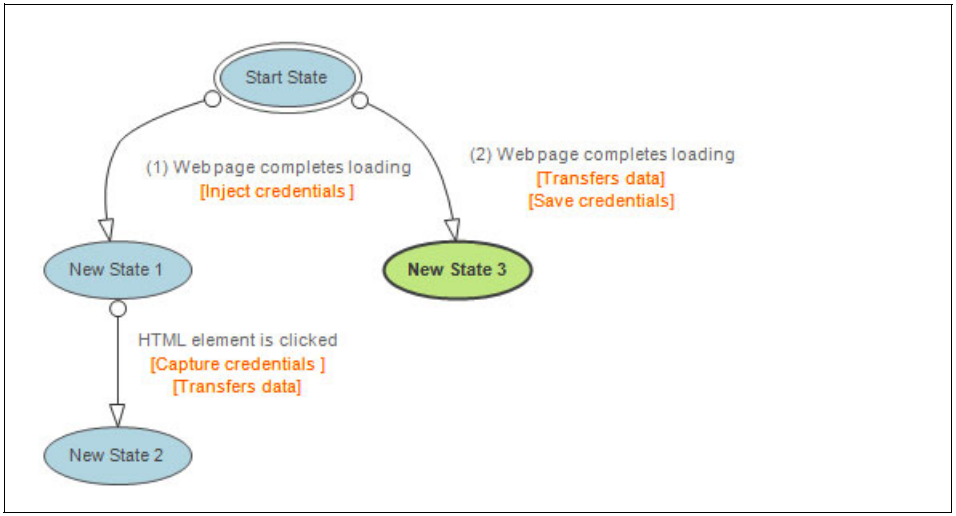

Figure B-8 Combined AccessProfile

Here, the first web page completes loading trigger handles, the login screen, and data capture, while the second web page completes loading trigger handles the saving of the credential. The use of global data between these two branches remains the same as Figure B-8.

The first instance of this AccessProfile is loaded for the login screen. It sets the global account-data for capture, and it is destroyed along with the containing frame. The second instance of the same AccessProfile has the second web page complete loading trigger fire when it sees the successful login page.

Solution 2

Another mechanism that does not involve the use of global bags is to write an AccessProfile for the main document with frame-aware signatures that are used to extract the necessary user name and password data from the child documents and to monitor the click of the Login button.

The signatures created by AccessStudio Signature Generator assume that the AccessProfile is meant to be written for the same document object as the element where it is dragged and dropped. Example B-22 shows the signature for the Login button in the example shown in Figure B-8 on page 338.

Example: B-22 Sample signature

```
/descendent::html[@tag_name="input" and @name="" and @type="button" and
@value="Login" and @id="submit"]
```

With this signature, the Login button is only in document object 2. Other document objects (1 and 3) search for this button within their web pages and do not find it.

If we prepend this signature with the following signature, we get the signature shown in Example B-23:

```
/child::html/ parent::frame/descendent::document[@url#".*f.html.*"]
```

Example: B-23 Sample signature

```
/child::html/
parent::frame/descendent::document[@url#".*f.html*"]/descendent::html[@
tag_name="input" and @name="" and @type="button" and @value="Login" and
@id="submit"]
```

This signature can access the Login button from all of the documents currently loaded in that browser tab.

Implementation: The directive `/child::html/parent::frame` looks arbitrary and, unfortunately, it is. It is a leftover of an older signature mechanism that refers to a hypothetical parent to all of the documents in the browser tab. It cannot be removed because of backward-compatibility restrictions. In a future release, the Observer provides a more intuitive directive to refer to all of the documents, but it continues to support this directive, so it is safe to use this mechanism.

Similarly, the signatures of the username and password fields can be converted to the signatures shown in Example B-24 and Example B-25.

Example: B-24 Sample signature

```
/child::html/
parent::frame/descendent::document[@url#".*f.html.*"]/descendent::html[
@tag_name="input" and @name="" and @type="text" and @id="username"]
```

Example: B-25 Sample signature

```
/child::html/
parent::frame/descendent::document[@url#".*f.html.*"]/descendent::html[
@tag_name="input" and @name="" and @type="text" and @id="pwd"]
```

This approach allows writing the AccessProfile in context of document object 1 and not document object 2. Because document object 1 is still around when the navigation occurs to the success screen (unlike document object 2, which is destroyed), a web page completes loading trigger can be added to the AccessProfile of Document object 1, which handles the success screen and saves the credential when that trigger fires.

Fortunately, the web page completes loading trigger of the top-level document (document object 1) occurs after all the other child documents' web document complete events occur. You can reliably expect the child documents and their HTML elements to be present if you want to access them (such as for injection, as shown in Figure B-9 on page 341) in the webpage completes loading trigger of the top-level document.

The AccessProfile site signatures therefore must match the top-level document. Example B-26 shows the signature.

Example: B-26 Sample signature

```
/child::web[@domain="localhost" and @protocol="http" and
@url~".*a.html"]
```

The AccessProfile structure looks like Figure B-9.

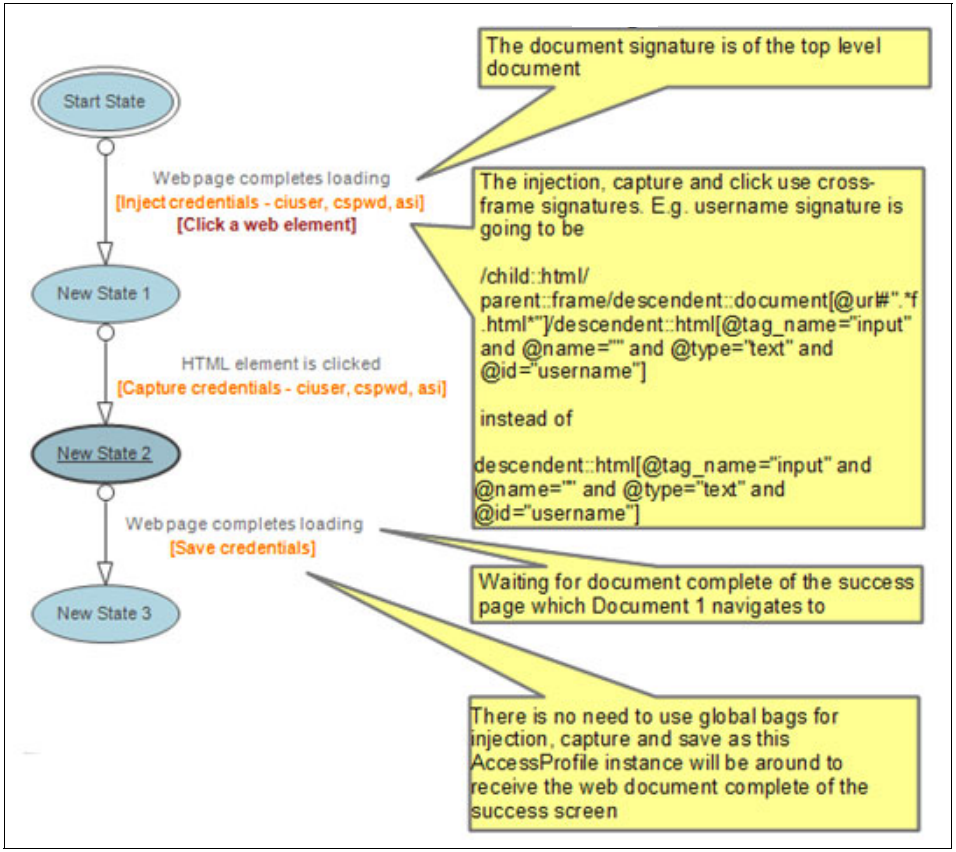

Figure B-9 The AccessProfile structure

Solution 1 versus Solution 2

Solution 2 (by using cross-frame signatures) simplifies the AccessProfile; however, it is written with the assumption that document object 1 is the top-level document. Certain websites reuse the login page document (document object 2) by embedding it in another document, as in Figure B-9, and in other scenarios, by showing it as the top-level document.

So, the user can see both of the following in several workflows for the web application. Alternatively, the AccessProfile, which relied on document object 1, can no longer be loaded in this independent login page.

Solution 1 (using global bags) does not have this problem, because it does not care where the login screen is hosted (independently or under a frame). It does

however have to work with global bags that need to be cleared at an appropriate time to avoid reusing stale data, and other problems.

The SSO/automation workflow needs to access fields from several documents. Consider the web pages shown in Example B-27, Example B-28, and Example B-29.

Example: B-27 h.html (contains two frames)

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<script type="text/javascript">
function clicked() {
window.parent.document.URL="e.html";
}
</script>
This is is the main web page - H
<br /><br />
<iframe src="g.html" height="100" width="300"></iframe>
<br /><br />
<iframe src="f.html" height="100" width="300"></iframe>
<br /><br />
<input type="button" id="submit" onclick="clicked();" value="Login"/>
</body>
</html>
```

Example: B-28 g.html (contains the username field)

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
This is is the contained web page - G
<br /><br />
User: <input type="text" id="username" />
</body>
</html>
```

Example: B-29 f.html (contains the password field)

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```
<title></title>
</head>
<body>
This is is the contained web page - F
<br /><br />
Pwd: <input type="text" id="pwd" />
</body>
</html>
```

When loaded, `h.html` looks like Figure B-10.

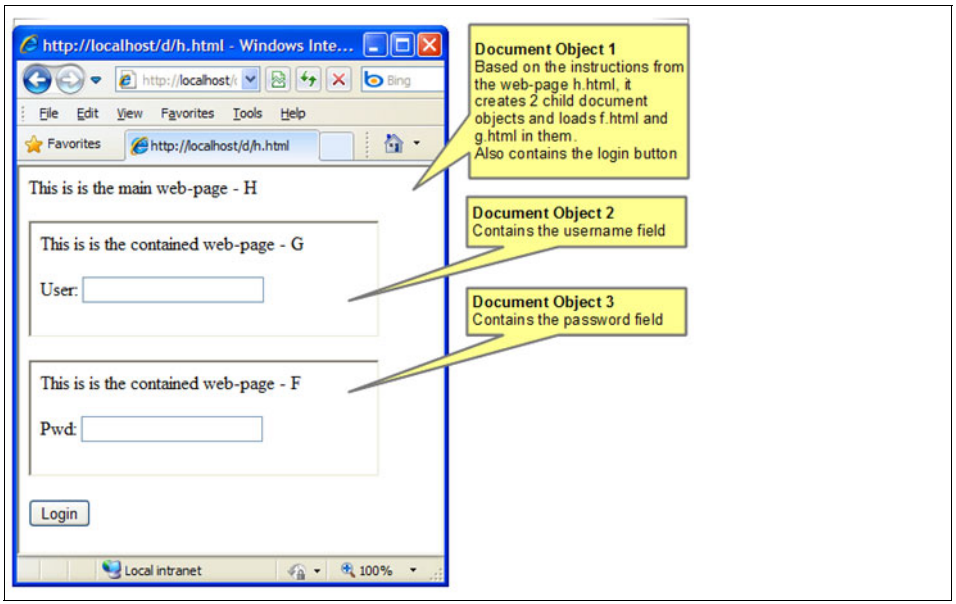

Figure B-10 h.html

To successfully capture the credentials, the AccessProfile must retrieve data from both document object 2 and document object 3. In this case, the solution is to use a similar technique described in “Solution 2” on page 339.

The idea is to use cross-frame signatures to refer to HTML elements that are not contained in the same document. It is easiest to write the AccessProfile for the top-level document object (document object 1) and use cross-frame signatures to get data and monitor events for elements not in the same frame.

The AccessProfile thus looks like the AccessProfile for solution 2, case 2, with the difference that the signature for the login screen stays as the following example, because it is in the same document as the AccessProfile instance:

```
/descendent::html[@tag_name="input" and @name="" and @type="button" and
@value="Login" and @id="submit"]
```

The username signature changes to the username signature shown in Example B-30.

Example: B-30 Sample username signature

```
/child::html/parent::frame/descendent::document[@url#".*g.html.*"]/desc
endent::html[@tag_name="input" and @name="" and @type="text" and
@id="username"]
```

The password signature changes to the password signature shown in Example B-31.

Example: B-31 Sample password signature

```
/child::html/parent::frame/descendent::document[@url#".*f.html.*"]/desc
endent::html[@tag_name="input" and @name="" and @type="text" and
@id="pwd"]
```

The `url` attribute of the document is the attribute that differentiates between `f.html` and `g.html`.

Frames support in V8.1

Tivoli Access Manager for Enterprise Single Sign-On V8.1 introduced the concept of a state-machine-id that helped track the lifetime of an AccessProfile instance. This state-machine-id view can be turned on by adding the registry entry shown in Example B-32.

Example: B-32 Registry key entry

```
Key: HKEY_LOCAL_MACHINE\SOFTWARE\Encentuate\AccessStudio
Name: ShowStateEngineIds
Type: DWORD
Value: 1
```

This registry key entry displays the state-machine-id next to every AccessStudio log. By using this log, it is possible to track the state-transitions, triggers fired,

and action executed against the correct state-machine when multiple state-machines are being loaded because of frames.

It is also possible to log the current URL for which a state-machine is loaded by inserting a dummy Data Transfer action with the following parameters:

- From:

```
Type = Web control
Signature = /child::html[@tag_name="BODY"]/@url
```

- To:

```
Type = Property store item
Name = var_dummy_url
```

This action creates a log similar to this log:

```
[State Machine Id - 8] Action: Transfers data. Property var_dummy_url is
set to 'http://localhost/d/h.html'
```

This action can help isolate the relevant state-machine for the URL of interest.

Frames support in V8.2

There are no changes in IBM Security Access Manager for Enterprise Single Sign-On V8.2 in the way that the Observer internally handle frames. However, the concept of the Document is now visible throughout AccessStudio and the Observer logs.

Every browser process log in AccessStudio now shows a Document and State Machine ID. It is easier to track the number of documents (and the frames) created for a page, the lifetime of AccessProfile instances loaded for it, and the URL associated with it at a specific instance.

Differences between Firefox and Internet Explorer AccessProfiles

In general, an AccessProfile written for a website in IE also works for Firefox and vice versa. Because all browsers handle HTML and JavaScript slightly differently, the server-side code that generates the content (ASP, JSP, or CGI) can return a different version of HTML to the browser based on its type. This difference might, in rare cases, invalidate a signature (because the attribute or the hierarchy of HTML elements might not be the same between the two browsers).

We suggest that you test the AccessProfile in both of the browsers before you deploy it in a mixed browser environment.

Several of the HTML element methods and available properties also differ between the two browsers. This difference is not much of an issue with the signatures. However, if the HTML Document is accessed directly by using an AccessProfile plug-in (by using the runtime.`GetHTMLDocument() call`), the way that the Document object is used by the plug-in script might need to differ for IE and Firefox. For example, the ID attribute value is case-insensitive in IE 7.0 and earlier, but not in Firefox and IE8 and later. Because this issue is not an AccessProfile-specific issue, a web search on browser differences while writing JavaScript directly can help reveal other areas of incompatibility.

Common issues

In this section, we describe common issues.

Slowness due to multiple auto-learn loads

In this section, we describe slowness due to multiple auto-learn loads.

Symptom

Navigation to a website that does not have any AccessProfile defined for it slowed noticeably after installing IBM Security Access Manager for Enterprise Single Sign-On.

Observations

Many (more than 20 or so) auto-learn AccessProfile loads happen while the web page loads in AccessStudio logs.

Probable cause

This issue might occur because the web page uses many frames or internally performs multiple navigations before it settles to the final page, each of which loads the auto-learn AccessProfile.

Resolution

Create two conflicting dummy AccessProfiles (the right site signature and one start state with no triggers) for the URLs seen in Observer logs for which the auto-learn AccessProfile is being loaded while the web page that exhibits slowdown is loaded.

Look for the following line in the log to determine the URL in Observer logs:

```
[CEnBrowserListener::DocumentComplete] Analyzing document URL is <URL>
```

Follow a few lines later in the log with this line:

```
[::GetApplicationObject] The AccessProfile id is
sso_site_web_auto_learn
```

Typically, these URLs have a common domain or a path that can be used to create a common site signature. If not, an AccessProfile allows specifying multiple site signatures. Either way, you do not need to create more than one set of conflicting AccessProfiles that can take care of all the cases where a deliberate conflict must be created.

In V8.2, the URL used to load an AccessProfile is evident from AccessStudio logs.

In rarer cases, where the client is not interested in the auto-learn facility, it is simpler to turn it off by using the `pid_sso_auto_learn_enabled` AccessProfile.

Further explanation

The auto-learn AccessProfile is loaded for each web page where no explicit AccessProfile is defined. If a web page has multiple navigations before it is displayed, the cumulative overhead of fetching an AccessProfile for each navigation adds up, causing the delay.

An AccessProfile is fetched in two steps. First, its ID is resolved based on the information provided by the agent to DataProvider. Then, the AccessProfile object is fetched by the agent by passing DataProvider the ID it retrieved. The latter step takes more time.

By creating a deliberate conflict, the agent is informed that multiple AccessProfiles were found for the web page whose details it passed to the DataProvider. When the agent receives this error, it does not try to retrieve any AccessProfile object from the DataProvider, which savings time and accelerates the page load.

No injection because an HTML element was not found

An error occurs when there is no injection because no HTML element was found.

Symptom

Injection does not occur even though the capture works fine.

Observations

AccessStudio logs show that an HTML element was not found, but when checked with the signature generator, the element highlights without any error.

Probable cause

This problem is a timing issue. A typical web AccessProfile tries to inject and click immediately after the web page completes loading trigger fires. However, when the user creates the signature of the various elements through the signature generator, other events in the web page might occur earlier that change the element (or the properties of its ancestors).

The signature generated at this later time in the lifetime of the page is invalid immediately after the web page completes loading trigger when injection and click are performed.

Resolution

Figure B-11 shows a typical web AccessProfile.

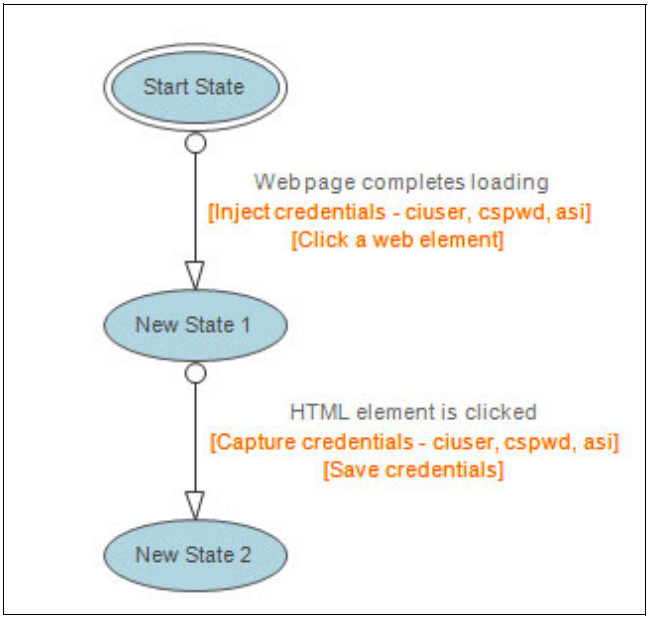

Figure B-11 Typical AccessProfile

The first step is to convert this AccessProfile, as shown in Figure B-12 on page 349.

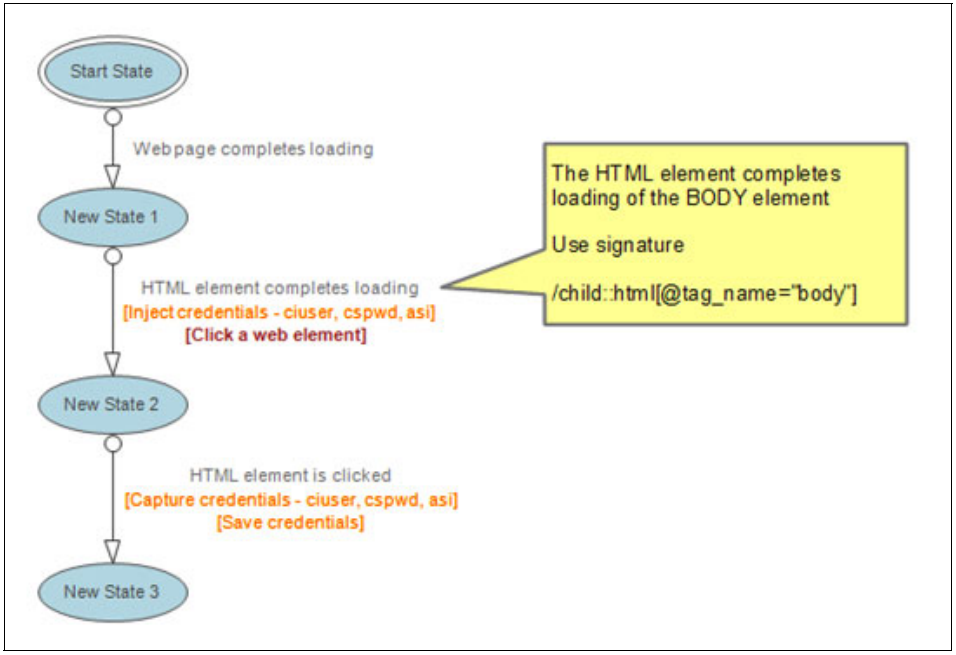

Figure B-12 Converting the AccessProfile

This conversion delays the injection and click until after the BODY element onload script (if any) fires. Because the signature in the AccessProfile was also created after the onload script executed, it increases the chances that the signature generated matches the state of the HTML elements at this time.

If AccessStudio logs show the HTML element not found error for the user name, password, or Submit button, further modify the AccessProfile (assume in this example that the user name input control was the INPUT control not found). See Figure B-13 on page 350.

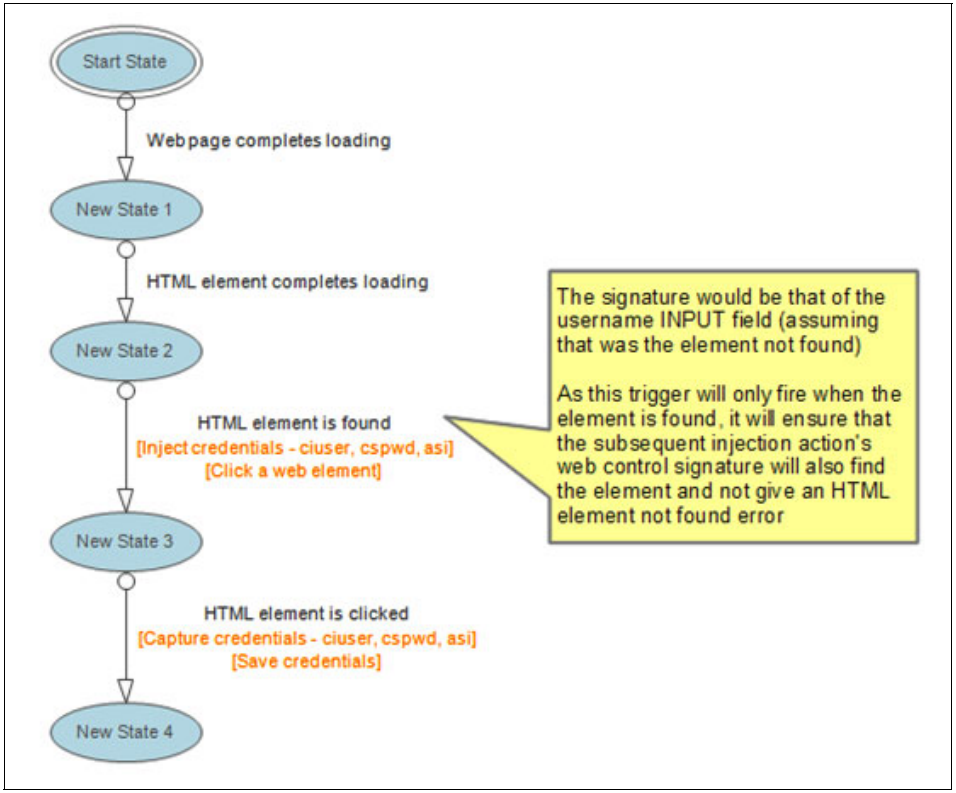

Figure B-13 With HTML element is found trigger

Further explanation

Actions, such as clicking an HTML element or injection into a web control, and triggers, such as waiting for a click of an HTML control, assume that the application is in a state when the control described by the signature is already present. These triggers, actions, and web controls do not wait for the control to become available later, they check for its presence one time only when they execute and, if not found, fail with a log in AccessStudio.

This design ensures that Observer does not degrade the system performance. For every trigger, action, or web control, extra resources are consumed to wait for the element to be found.

Instead, Observer provides triggers, such as HTML element completes loading, which help track the state of the application closely. In rare cases, where the only choice is to wait for an element to be visible before any trigger or an action can be performed on it, it provides an explicit HTML element found trigger (also the same logic for window found trigger for the Windows application).

Before the HTML element was found trigger was available in V8.1, the AccessProfile writer used a Fire after specified time trigger to wait for some time before injection (and hope the controls are all present by then). The use of fire after a specified time trigger is discouraged, because the necessary timeout values vary from machine to machine (based on load and processing power).

No injection and no HTML element not found error

In this section, we describe the `no injection` and `no HTML element not found` error.

Symptom

Injection does not occur even though the capture works fine.

Observations

AccessStudio logs show that the injection action fetched the credentials from the Wallet properly. It shows the account data bag that is populated with the correct user name and authentication service ID. There are no HTML element not found errors during injection.

Possible cause

The injection might happen, but then the INPUT HTML elements where injection was performed were cleared by another JavaScript in the web page. This situation typically occurs in the OnLoad event handler of the BODY element, which might be clearing the INPUT boxes.

Resolution

Delay the injection until after the OnLoad event of the BODY element occurs. Use the HTML element completes loading to wait for the BODY element to be loaded before injecting. See Figure B-14 on page 352.

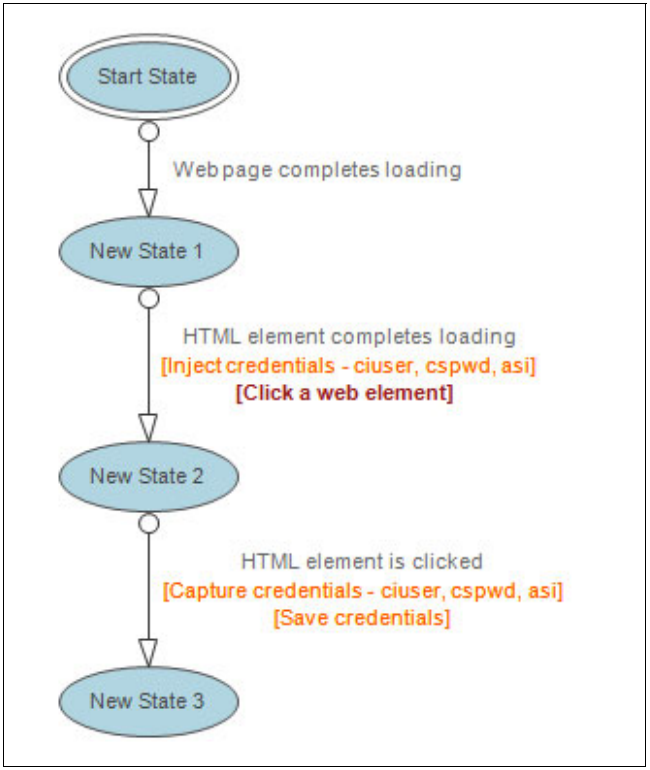

Figure B-14 Using onload event to delay injection

Further explanation

The resolution is the same as the first resolution tried in the case where there was an HTML element not found error in the AccessStudio logs. In that scenario, the HTML element properties were being modified so that the signature of the element was not matching before the OnLoad event.

In this case, the OnLoad event does not modify any properties of the HTML element that affect signature evaluation (it was still getting the right element). The OnLoad event was clearing the contents of the INPUT element. Any injection done before this event was being erased. Afterward, it appeared that no injection happened.

Captured credentials are obfuscated when saved

In this section, we describe the captured credentials are obfuscated when saved error.

Symptom

The AccessProfile is capturing obfuscated credentials, such as the following string: “*****”.

Observations

If the user name is being captured obfuscated, the web page converts the entered user name partially or entirely to the following string, “********”, when the user selects another field (typically the password) after entering a user name.

Possible causes

The website uses a JavaScript to obfuscate the user name or password display for security reasons whenever the INPUT control loses focus.

Resolution

Use the HTML element lost focus trigger to retrieve the credentials before they get obfuscated by the JavaScript. The AccessProfile looks like Figure B-15.

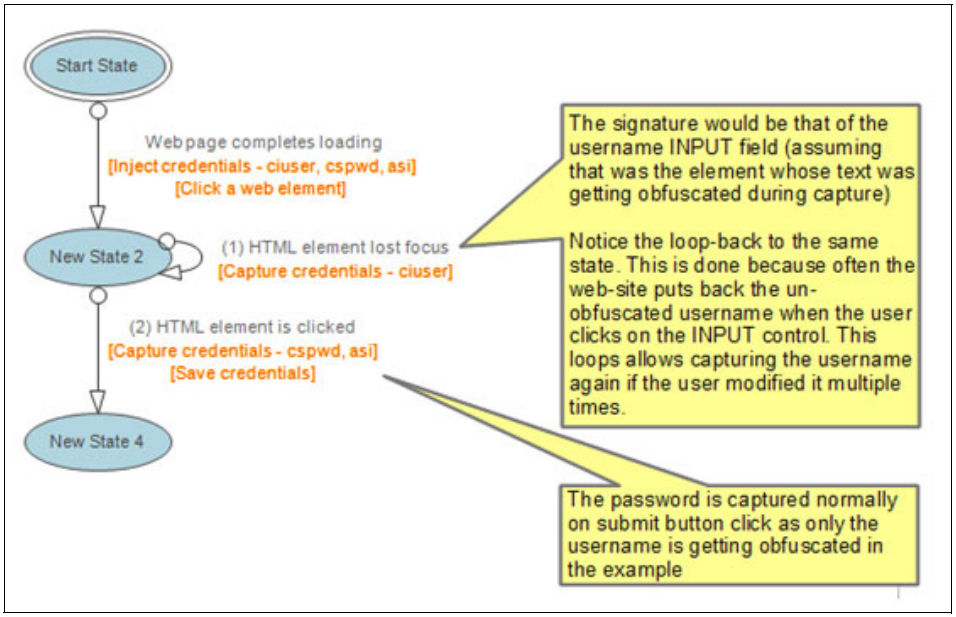

Figure B-15 AccessProfile with HTML element lost focus trigger

In certain cases, the web page might obfuscate the user name as it is typed. In this case, the web page keeps the actual user name in a hidden HTML element, typically another INPUT element with type set to HIDDEN.

The exact mechanism can be determined only by examining the HTML and the embedded JavaScript in the web page. One option is to use the Run a VBScript or JavaScript action under the HTML element clicked trigger to access the HTML Document Object Model (DOM) directly and retrieve the value.

Example B-33 shows a sample VBScript.

Example: B-33 Sample VBScript

```
' Get the current HTML Document
set doc=runtime.GetHTMLDocument()

' Assuming the userid was stored in a different form – frmPoster's
hidden ssn 'field
userid=doc.frmPoster.ssn.value
set propcon=runtime.GetPropertiesContainer()

'Assumes that the capture bag is already created
propcon.SetAccDataItem("default_capture_bag", "aditi_ciuser", userid)
```

This VBScript retrieves the value from an HTML page, which looks like Example B-34 and uses the frmPoster's hidden input controls to temporarily keep the username while the INPUT control, which shows the username, is obfuscated.

Example: B-34 Sample HTML

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<form name="frmPoster" id="frmPoster" method="POST" action="abc\def" >
<input type="hidden" maxlength="15" name="ssn">
<input type="hidden" maxlength="12" name="pin">
</form>
</body>
</html>
```

Another option is to set the signature of the web control in the capture field to refer to the HIDDEN field, as shown in Example B-35 on page 355. It is impossible to create this signature by the Signature Generator's drag and drop facility, because it is a hidden facility and there is nothing visible on which to drag and drop the control selector.

Example: B-35 Sample signature

```
/descendent::html[@tag_name="form" and
@name=""frmPoster]/descendent::html[@tag_name="input" and @name="ssn"
and @type="hidden"]
```

Further explanation

HTML elements provide an OnBlur event that is called when the control loses focus and can be handled by using the HTML Element lost focus trigger. It is possible for a web page to trap this event and write JavaScript for it, which transfers the original text in the INPUT element to a hidden HTML and modifies the display text.

Save action not firing

In this section, we describe the save action not firing error.

Symptom

Save does not happen for a validating AccessProfile.

Observations

The trigger that contains the save action (typically, a web page completes loading trigger of the validation page) never fires.

When the click happens, the AccessStudio logs show the correct capture of the credentials. Then, the AccessStudio logs unexpectedly show the auto-learn AccessProfile getting loaded and no further logs show for the current AccessProfile instance (identified by the State Machine Id in the logs).

If no auto-learn AccessProfile is present (happens in AccessStudio Test mode when there is no auto-learn AccessProfile available), instead of the auto-learn AccessProfile loading log, the following information is shown:

```
AccessProfile: <AccessProfileId>. Unloaded because it did not handle
the URL: <URL>
```

In both cases, the URL used to unload the current AccessProfile instance and load the auto-learn AccessProfile can be seen in the Observer logs, as well.

Possible cause

This error probably happens because the state-machine was in a state that did not handle the intermediate navigation to the URL, and the site signatures of the current AccessProfile did not match this URL either. This situation caused the

unloading of the AccessProfile instance, and the necessary web page completes loading trigger that was to detect the presence of the validation screen never fired.

This error can also happen because the web application used frames in a way that the document object under which the save and capture were supposed to occur was destroyed. In this case, use the techniques outlined in “Frames and the web browser document object” on page 330.

Resolution

Either put in a webpage completes loading trigger for the URL, which causes the AccessProfile unload in the state with the web page completes loading trigger that contains the save action, or add or modify the site signatures so that this URL matches this AccessProfile.

In both cases, the AccessProfile instance no longer unloads. If there were multiple navigations that happened that caused the AccessProfile instance to unload, this addition of the webpage completes loading trigger or the site signature modification might need to occur multiple times.

Figure B-16 describes the webpage completes loading trigger process.

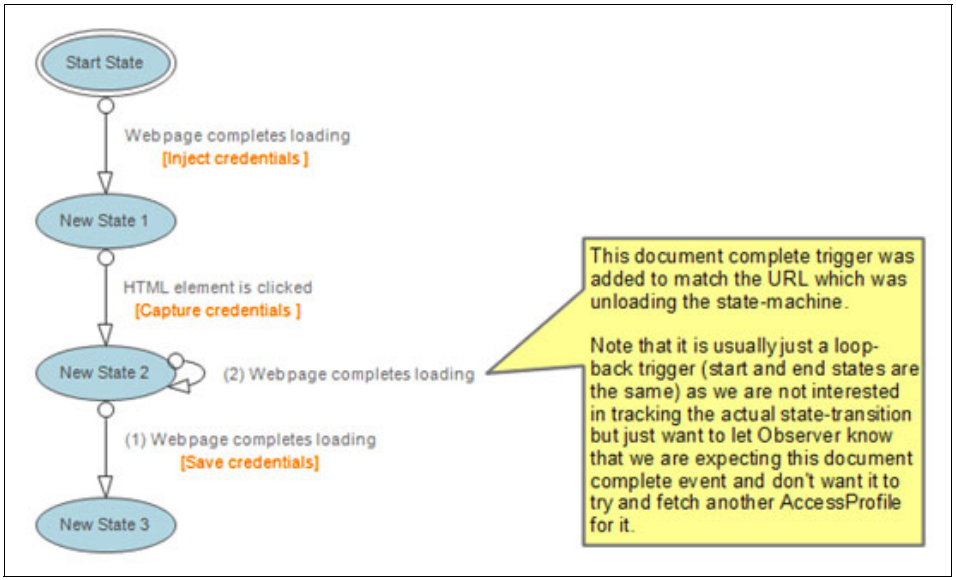

Figure B-16 Webpage completes loading trigger to prevent AccessProfile unloading

Further explanation

For further details about the lifetime management of an AccessProfile instance, see “Document complete event and the Observer” on page 322.

Slow injection or clicking when using fire after some time trigger

In this section, we describe the slow injection or clicking when using fire after some time trigger error.

Symptom

Slow credential injection or a button click in a web page.

Observations

It takes a few seconds for the injection into web controls to occur when the injection is being done under a fire after specified time trigger. Similarly, clicking an HTML element is slower, as well.

Possible cause

This error is an implementation limitation on the Observer side. Internally, the fire after specified time runs its actions under a different thread. Windows OS does additional processing when the HTML Document is accessed from a different thread than the one that created it.

This error is the same reason why AccessStudio signature creation and highlight functions are slower than a typical injection to the same web control.

Resolution

In V8.1, avoid injection under a fire after specified time trigger and after a wait for some time action. Instead, use the HTML element found trigger if the delay was necessary for the HTML element to be created. This limitation is removed in V8.2.

Further explanation

The HTML element found trigger runs in the same thread as the one that owns the Document object. It is also a better way to wait for an HTML element to be created rather than by using a fire after specified time trigger. A fire after specified time trigger can give unpredictable results, because different machines and browsers might take different amounts of time to create an HTML element.

No AccessProfile loading for a non-browser hosted web page

In this section, we describe the no AccessProfile loading for a non-browser hosted web page error.

Symptom

No AccessProfile is being loaded for the web page.

Observations

The web page is being displayed in a non-browser application.

Possible cause

The web SSO agent is not loaded for non-browser applications, and therefore, there is no automation or SSO.

Resolution

Use the Start Installing BHO action for the window that needs to have automation and monitoring support enabled. A separate web AccessProfile is written for the web page that is hosted in this application, the same as though this web page were hosted in a normal browser.

Figure B-17 describes the Start Installing BHO action.

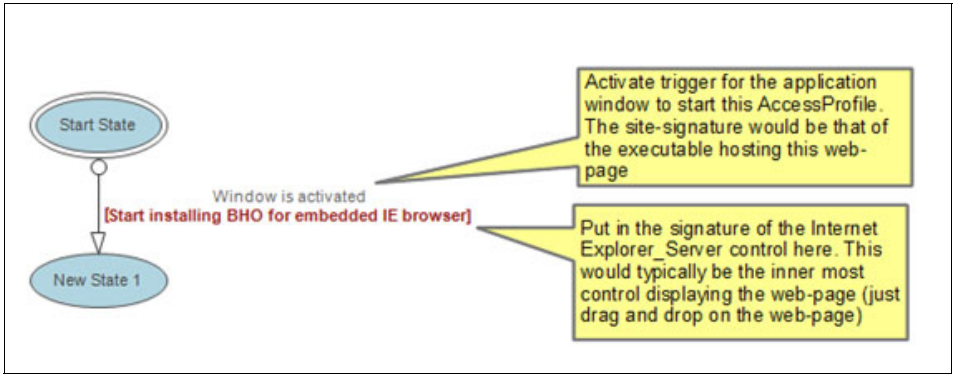

Figure B-17 Use start installing BHO action

Further explanation

Internet Explorer provides a reusable browser window that can be embedded in other applications, such as Word, PowerPoint, and other applications that at times embed the browser window within their own window.

Observer uses a plug-in mechanism in Internet Explorer called *Browser Helper Object* (BHO) to load itself in its process and monitor the web content. If the Internet Explorer component for displaying the web page is embedded in another application, the BHO does not load for that application and thus there is no SSO. The start installing BHO action instructs the Internet Explorer component to load the BHO and enables the usual web monitoring and automation.

This mechanism to force the application to load the BHO works only if it uses the Internet Explorer_Server component that comes with Internet Explorer. For this reason, identify the Internet Explorer_Server control in the Start installing BHO action.

There is a corresponding Stop installing BHO action, as well. There is no need to call it, but in rare cases the application has multiple embedded browser windows and the SSO support needs to be available for only one of those windows. Then, the Stop installing BHO action can be used after the BHO is installed by the Start installing BHO action for the window specified by its signature.

In the remainder of this appendix, we provide two scenarios for using *advanced profiling techniques*. We show specific examples of how to integrate these applications, by using XML Path Language (*XPath*), a language that facilitates XML document navigation, and by using Microsoft Visual Basic Script (VBScript).

Use case

Specific applications cannot be integrated with the AccessStudio Assistant only. Certain applications might use dynamic IDs. That is, the control ID for the password field can differ with every restart of the applications. Other applications might seem difficult to integrate at first, because the control IDs for the user name and password cannot be distinguished from each other with standard methods. Let us describe a *use case* of non-web-based advanced profiling to show this discipline.

Use case: Start a program from the command line

This use case demonstrates how to start an application from the command line of an AccessProfile:

- We implement an AccessProfile that captures the user name and password.
- IBM Security Access Manager for Enterprise Single Sign-On starts an application and injects the user name and password as additional command-line parameters.

Our example includes a fictitious program named “Cardio Client” that can be started through the command line, for example:

```
C:\Program Files\Cardio\Cardio.exe /u:<user>, <pwd>
```

The complete AccessProfile is shown in Figure B-18. We now describe the details.

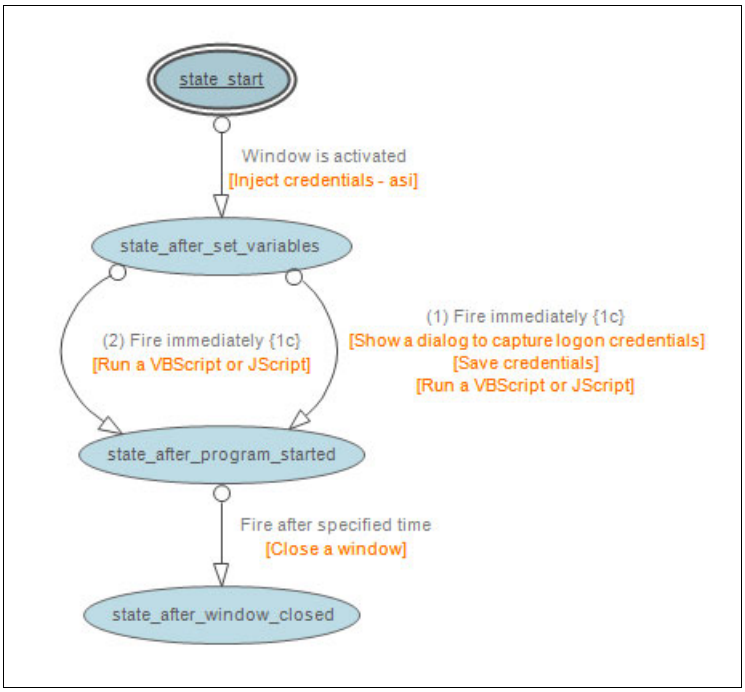

Figure B-18 AccessProfile overview

To start a program from a command line:

1. First, we must decide when to start this command line. In this use case, we create a short VBScript to set a trigger. The user starts it by double-clicking a desktop icon. IBM Security Access Manager for Enterprise Single Sign-On detects this action as we define the correct signature in an AccessProfile.

   Let us assume the application is named “Cardio Client” and so we create the following VBScript:

   ```
   Const wshOK = 64
   Set objShell = CreateObject("Wscript.Shell")
   intReturn = objShell.Popup("Starting cardio", 0, "Cardio Client", wshOK)
   ```

   The number `0` (zero) in the last line specifies that the window automatically closes after 0 seconds, which means that it never closes. The text `"Starting`

cardio" bears no consequence in our case, it defines a window message only and can be seen in Figure B-20. Only the text "Cardio Client", which defines the title of the pop-up window, triggers in our example.

We then create a shortcut to this script on the desktop, as shown in Figure B-19.

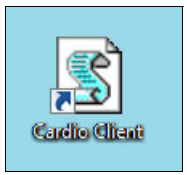

Figure B-19 The VBScript as desktop icon

Double-clicking this shortcut opens a dialog, as shown in Figure B-20.

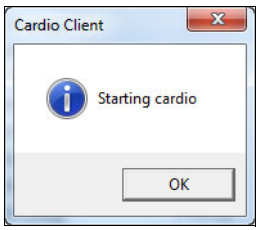

Figure B-20 The VBScript pop-up window

This dialog is our trigger for the AccessProfile.

2. In the AccessStudio, we must define a signature for the VBScript on the General Properties tab. It includes the executable:

   ```
   /child::exe[@exe_name="wscript.exe"]
   ```

3. We must distinguish our VBscript from any other script, so the first trigger in the AccessProfile, as shown in Figure B-21 on page 362, includes more details:

   ```
   /child::wnd[@title="Cardio Client" and @class_name="#32770"]
   ```

   We match our example VBScript only, because we specifically trigger for the title **"**`Cardio Client`**"**. The **"**`#32770`**"** is a standard window class.

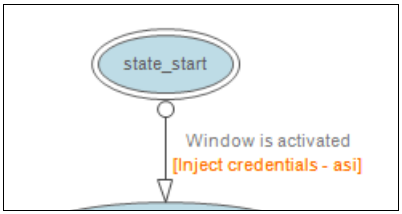

Figure B-21 Trigger - (When a) window is activated

4. The trigger, Fires immediately, in the next state (see Figure B-23 on page 363) includes the condition that is shown in Figure B-22 on page 363, NO_ACCOUNT_DATA_FOUND=1, which means that the credentials bag for this AccessProfile is empty. The next actions are to capture the user credentials and to save them. The last action includes a VBScript, built into AccessProfile, and its security context. It starts “Cardio Client” from a command line on behalf of the user and includes the credentials user name and password as parameters to sign on the user to the “Cardio Client”. We describe this script later.

5. The triggers in Figure B-23 on page 363 never trigger if we do not enter the action `Auto-fills user credentials`, as shown in Figure B-21, because the property NO_ACCOUNT_DATA_FOUND is not defined yet. We create a *dummy action* that contains no injection fields at all. However, it initializes the property NO_ACCOUNT_DATA_FOUND and sets it to `1`.

   If this action is not set, the `aa_observer.log` includes this line:

   ```
   {WScript.exe}  [CObsTrigger::TestProperties]  Property not in bag,
   id = NO_ACCOUNT_DATA_FOUND
   ```

   Another essential task is to define an authentication service in this action, in our example, the direct authentication service auth_CardioClient. The text “asi” in the action “Auto-fills user credentials - asi” shows that we set an authentication service.

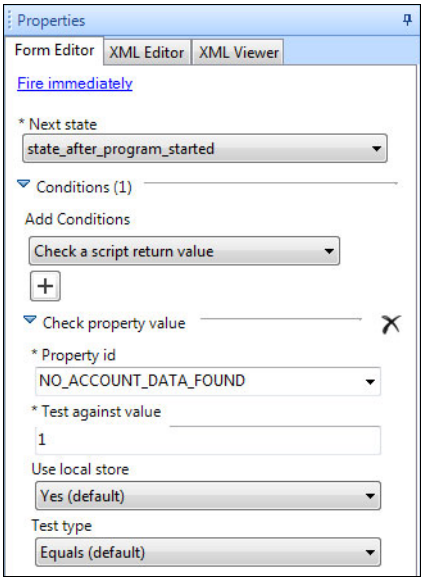

Figure B-22 Condition NO_ACCOUNT_DATA_FOUND

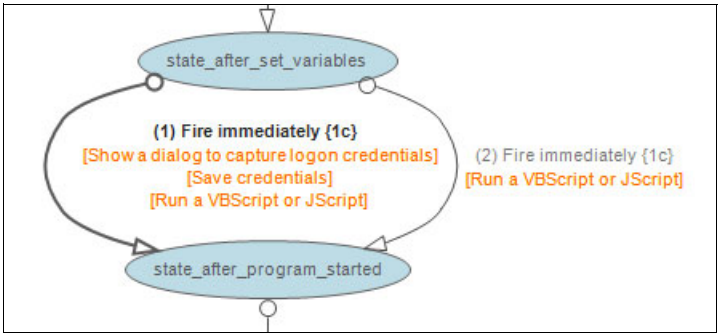

Figure B-23 Trigger When no account data exists

6. The next time that the user starts the VBScript, the property NO_ACCOUNT_DATA_FOUND is no longer true. The other trigger, as shown in Figure B-24 on page 364, fires.

   Here, we test the condition NO_ACCOUNT_DATA_FOUND=0.

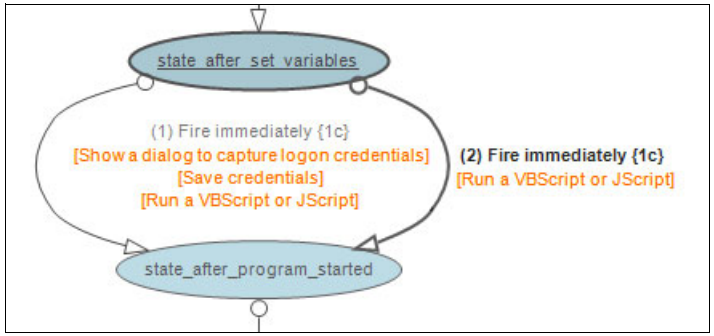

Figure B-24 Trigger When account data exists

7. We define the action “Shows a dialog to capture user’s logon credentials”. We chose the signature of the OK Button, including the string Cardio Client:

   ```
   /child::wnd[@title="Cardio Client" and
   @class_name="#32770"]/child::wnd[@class_name="Button" and
   @ctrl_id=2]
   ```

   The authentication service is auth_CardioClient, and credentials to be saved are defined in Advanced Options this way:

   ```
   adt_ciuser_cspwd (default)
   ```

8. The next action saves the user credentials. Alternatively, we also can configure the last action to store the captured credentials into the Wallet directly.

9. The next trigger, as shown in Figure B-25, includes the last action. It closes the VBScript pop-up window. This trigger also closes our AccessProfile pop-up window, which must run a VBScript to start the `"Cardio Client"` because this action is a child process. However, the separate `"Cardio Client"` process is not closed, because it is a separate process. We set this trigger to wait 2 seconds. The following signature is for this action:

   ```
   /child::wnd[@title="Cardio Client" and @class_name="#32770"]
   ```

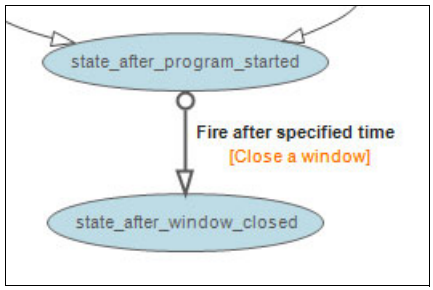

Figure B-25 Action to close the VBScript pop-up window

10. Finally, we look at the VBScript to start the "Cardio Client":

```
set pc = runtime.GetPropertiesContainer()
user = pc.GetAccDataItem("default_capture_bag","aditi_ciuser",1)
pwd = pc.GetAccDataItem("default_capture_bag","aditi_cspwd",1)

strCmd = "C:\Program Files\Cardio\Cardio.exe /u:" & user & "," & pwd

set wsShell = CreateObject ("WScript.Shell")
wsShell.run strCmd
set wsShell = nothing

‘MsgBox("user: " & user & "; pwd:" & pwd & "; strCmd: " &  strCmd)
```

First, we get the runtime variables of our AccessProfile and read user name (ciuser, case-insensitive user) and password (cspwd, case-sensitive password) out of the default capture bag into two variables.

Then, we build our command line, as shown at the beginning of this use case:

```
strCmd = "C:\Program Files\Cardio\Cardio.exe /u:" & user & "," & pwd
```

Finally, we execute the command to start the command line.

The next message box (MsgBox) is for debugging only and can be commented.

To finalize this use case, we attach the complete AccessProfile Cardio.eas in Example B-36.

Example: B-36 Complete AccessProfile Cardio.eas

```
<all_data>
  <application creation_timestamp=""
entity_id="scp-id-current-ims-017b02c7-7f23-4ca6-878c-16f8082208d8" entity_type="scp_ims"
is_deleted="0" update_timestamp="" _origin="file">
    <storage_template_id_value_pair>
      <id>app_CardioClient</id>
      <value>
        <name>CardioClient</name>
        <desc />
      </value>
    </storage_template_id_value_pair>
  </application>
  <authenticator creation_timestamp=""
entity_id="scp-id-current-ims-017b02c7-7f23-4ca6-878c-16f8082208d8" entity_type="scp_ims"
is_deleted="0" update_timestamp="" _origin="file">
    <storage_template_id_value_pair>
      <id>auth_CardioClient</id>
      <value>
        <disp_name>CardioClient</disp_name>
        <description />
        <monikers type="mt_injection" />
        <monikers type="mt_capture" />
        <account_data_template_id>adt_cspwd</account_data_template_id>
      </value>
    </storage_template_id_value_pair>
  </authenticator>
```

```
  <sso_site creation_timestamp="" entity_id="scp-id-current-ims-017b02c7-7f23-4ca6-878c-16f8082208d8"
entity_type="scp_ims" is_deleted="0" update_timestamp="" _origin="file">
    <storage_template_id_value_pair>
      <id>profile_CardioClient</id>
      <meta>
        <as_meta>
          <sso_wizard_info>
            <is_wizard_gen>1</is_wizard_gen>
          </sso_wizard_info>
        </as_meta>
      </meta>
      <value>
        <site_signatures>
          <site_signature>/child::exe[@exe_name="wscript.exe"]</site_signature>
        </site_signatures>
        <site_info>
          <application_id>app_CardioClient</application_id>
        </site_info>
        <sso_support>
          <state_engine_sso_support xmlns:msxsl="urn:schemas-microsoft-com:xslt">
            <states>
              <state id="1031410107668424614101071306601181414110121 2" is_begin_state="1">
                <state_name>state_start</state_name>
                <triggers>
                  <trigger>
                    <wnd_activate_trigger>
                      <signature>/child::wnd[@title="Cardio Client" and
@class_name="#32770"]</signature>
                      <wnd_match_lparam type="title_text">Encentuate AccessAgent</wnd_match_lparam>
                      <wnd_match_lparam_type>6</wnd_match_lparam_type>
                      <actions>
                        <action>
                          <acc_data_inject_action>
                            <acc_data_bag id="default_injection_bag">
                            </acc_data_bag>
                            <action_id>81110451141401411940651057312573141110131102</action_id>
                            <auth_info>
                              <direct_auth_info>
                                <auth_id>auth_CardioClient</auth_id>
                              </direct_auth_info>
                            </auth_info>
                          </acc_data_inject_action>
                        </action>
                      </actions>
                      <next_state_id>9142921501214131394791411831555096114711111153</next_state_id>
                      <trigger_id>1126116261191515141138914571711171111139946</trigger_id>
                    </wnd_activate_trigger>
                  </trigger>
                </triggers>
              </state>
              <state id="914292150121413139479141183155509611471111153">
                <state_name>state_after_set_variables</state_name>
                <triggers>
                  <trigger>
                    <gen_null_trigger>
                      <next_state_id>153118710082002472129157359211153738010 3</next_state_id>
                      <actions>
                        <action>
                          <observer_dlg_capture_action>
                            <acc_data_bag id="default_capture_bag" />
                            <action_id>902515801270894459112271506425111201211 0</action_id>
                            <auth_info>
                              <direct_auth_info>
                                <auth_id>auth_CardioClient</auth_id>
                              </direct_auth_info>
                            </auth_info>
```

```
                          <signature>/child::wnd[@title="Cardio Client" and
@class_name="#32770"]/child::wnd[@class_name="Button" and @ctrl_id=2]</signature>
                        </observer_dlg_capture_action>
                      </action>
                      <action>
                        <acc_data_save_action>
                          <acc_data_bag_id>default_capture_bag</acc_data_bag_id>
                          <action_id>81101244661015141349391097170118186914536</action_id>
                        </acc_data_save_action>
                      </action>
                      <action>
                        <run_script_action>
                          <script language="VBScript"><![CDATA[set pc =
runtime.GetPropertiesContainer()
user = pc.GetAccDataItem("default_injection_bag","aditi_ciuser",1)
pwd = pc.GetAccDataItem("default_injection_bag","aditi_cspwd",1)
strCmd = "C:\Program Files\Cardio\Cardio.exe /u:" & user & "," & pwd
set wsShell = CreateObject ("WScript.Shell")
wsShell.run strCmd
set wsShell = nothing
'MsgBox("user: " & user & "; pwd:" & pwd & "; strCmd: " & strCmd)
]]></script>
                          <action_id>1149132913149914104151413961141410762132111491135</action_id>
                        </run_script_action>
                      </action>
                    </actions>
                    <trigger_id>12151336913114114410132911153101411715120101134</trigger_id>
                    <conditions>
                      <condition>
                        <test_property id="NO_ACCOUNT_DATA_FOUND">
                          <test_value>1</test_value>
                        </test_property>
                      </condition>
                    </conditions>
                  </gen_null_trigger>
                </trigger>
                <trigger>
                  <gen_null_trigger>
                    <next_state_id>153118710082002472129157359211153738010 3</next_state_id>
                    <actions>
                      <action>
                        <run_script_action>
                          <script language="VBScript"><![CDATA[set pc =
runtime.GetPropertiesContainer()
user = pc.GetAccDataItem("default_injection_bag","aditi_ciuser",1)
pwd = pc.GetAccDataItem("default_injection_bag","aditi_cspwd",1)
strCmd = "C:\Program Files\Cardio\Cardio.exe /u:" & user & "," & pwd
set wsShell = CreateObject ("WScript.Shell")
wsShell.run strCmd
set wsShell = nothing
'MsgBox("user: " & user & "; pwd:" & pwd & "; strCmd: " & strCmd)
]]></script>
                          <action_id>0887711155135842311118022113226107810313</action_id>
                        </run_script_action>
                      </action>
                    </actions>
                    <trigger_id>651114121410106071141213483141121553925112824</trigger_id>
                    <conditions>
                      <condition>
                        <test_property id="NO_ACCOUNT_DATA_FOUND">
                          <test_value>0</test_value>
                        </test_property>
                      </condition>
                    </conditions>
                  </gen_null_trigger>
                </trigger>
              </triggers>
```

```
            </state>
            <state id="153118710082002472129157359211153738010 3">
              <state_name>state_after_program_started</state_name>
              <triggers>
                <trigger>
                  <gen_time_out_trigger>
                    <next_state_id>76845101254131424105680451514948201311410</next_state_id>
                    <time_out>2</time_out>
                    <actions>
                      <action>
                        <wnd_close_window_action>
                          <signature>/child::wnd[@title="Cardio Client" and
@class_name="#32770"]</signature>
                          <action_id>69601371471312138476511131515139112866614111 16</action_id>
                        </wnd_close_window_action>
                      </action>
                    </actions>
                    <trigger_id>74101111432111512443111391341113282151421313111 0</trigger_id>
                  </gen_time_out_trigger>
                </trigger>
              </triggers>
            </state>
            <state id="76845101254131424105680451514948201311410">
              <state_name>state_after_window_closed</state_name>
              <triggers />
            </state>
          </states>
        </state_engine_sso_support>
      </sso_support>
    </value>
  </storage_template_id_value_pair>
 </sso_site>
</all_data>
```

Conclusion

In this appendix, we investigate how to integrate web-based applications into IBM Security Access Manager for Enterprise Single Sign-On by using the AccessProfile technology.

After looking at background information, we present the technologies around the Observer and Signatures, and how to auto-learn an AccessProfile and how to handle basic authentication. Then, we look at frames and the web browser document object and explain the differences between Firefox and Internet Explorer AccessProfiles.

Finally, we stop describing pure web-based application profiling and look at a common, realistic use case.

C

Configuring strong authentication

IBM Security Access Manager for Enterprise Single Sign-On automates sign-on and access to enterprise applications, eliminating the need to remember and manage user names and passwords. Users log on to IBM Security Access Manager for Enterprise Single Sign-On with a special user ID and password. Then, when they access their secured applications, the IBM Security Access Manager for Enterprise Single Sign-On agent injects their stored credentials automatically without requiring the users to enter them. IBM Security Access Manager for Enterprise Single Sign-On provides the usual features associated with password security, for example, password length and aging policy.

This appendix is based on a set of exercises that was produced for the European Tivoli Technical Conference 2010. It shows how to configure IBM Security Access Manager for Enterprise Single Sign-On to use additional or alternative methods of authentication when users log on to provide a greater degree of security (stronger authentication). This appendix includes the following sections:

- "Configuring authentication to use smart cards" on page 371
- "Configuring authentication to use radio frequency identification cards" on page 408
- "Strong authentication by using biometrics" on page 416

- “Configuring authentication for Mobile ActiveCode as a one-time password” on page 444

Why do these screen captures look different? Because this set of exercises was implemented by using a previous version of IBM Security Access Manager for Enterprise Single Sign-On, the screen captures show the previous version. Because the technical details apply to IBM Security Access Manager for Enterprise Single Sign-On 8.2, we decided to include this valuable appendix in the book.

Configuring authentication to use smart cards

This section explains how to configure an existing IBM Security Access Manager for Enterprise Single Sign-On environment to use smart cards as additional authentication factors.

This section includes the following topics:

- "Prerequisite environment" on page 371
- "Testing smart card compatibility" on page 372
- "Configuring the certificate authority" on page 374
- "Importing the CA root certificate to the HTTP Server truststore, part 1" on page 378
- "Importing the CA root certificate to the HTTP Server truststore, part 2" on page 381
- "Enabling two-way Secure Sockets Layer on the IBM HTTP Server" on page 390
- "Creating IMS Server policies for smart card use" on page 392
- Assigning the new template to the client workstation
- "Modifying the user default template to accept smart card authentication" on page 395
- "Issuing a certificate to a smart card" on page 397
- "Registering a smart card to a user" on page 405

Prerequisite environment

To run this exercise, you need the following resources. See the IBM Security Access Manager for Enterprise Single Sign-On product documentation for platform requirements and configuration instructions:

- Integrated Management System Server (IMS Server):
  - Microsoft Certificate Server
  - Internet Information Services
  - IBM Security Access Manager for Enterprise Single Sign-On IMS Server prerequisites:
    - IBM WebSphere Application Server
    - IBM HTTP Server
    - A supported database (for example, IBM DB2)

– Smart card middleware

Middleware: This scenario uses the Charismathics Smart Security Interface (CSSI).

- Client:
  - IBM Security Access Manager for Enterprise Single Sign-On AccessAgent
  - Smart card middleware
  - Initialized smart card and reader or USB token
  - Drivers for the reader or token
- Active Directory

  Domain that contains the computers and user accounts

Testing smart card compatibility

The Smart Card Compatibility Tool is supplied with IBM Security Access Manager for Enterprise Single Sign-On installation files. The tool is installed in the `SCardCompatTool` *`version`* directory.

You can test smart card compatibility in the following way:

1. Create a `mycsp.ini` configuration file that contains the details of the location of the smart card middleware driver, by using the supplied `example.ini` file for guidance.
2. Run the following command from the command line:

   `SCardCompatTool.exe -i mycsp.ini -o` *`output_file_name`*

   The following prompt appears:

   ```
   Insert the smart card that you wish to test. Press Enter to proceed.
   ```

3. Insert the smart card into the reader, and press Enter. Then, enter the PIN when prompted. Figure C-1 shows the tests.

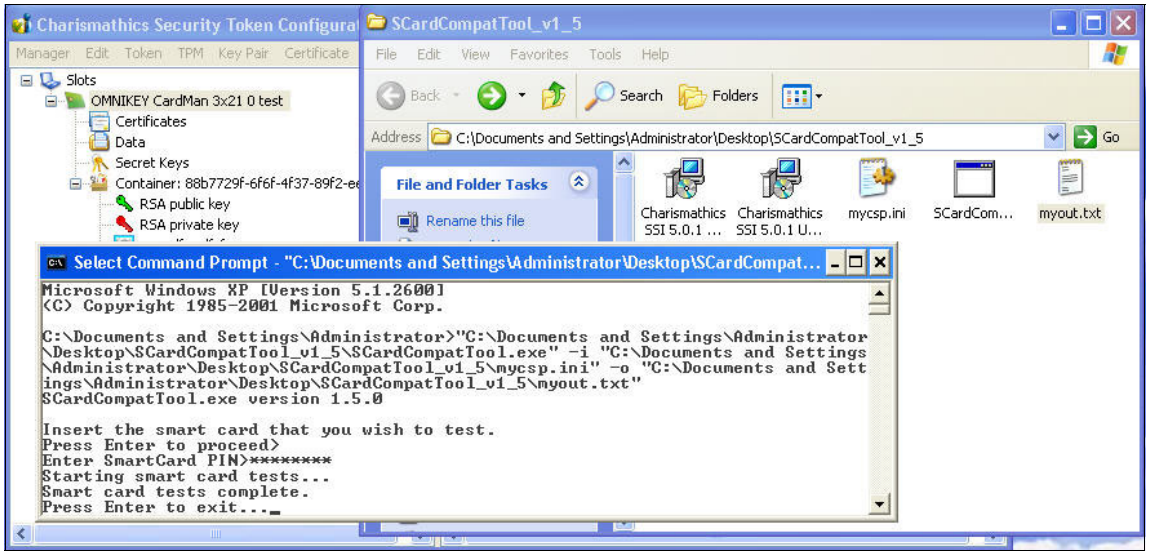

Figure C-1 Smart card compatibility tool

4. Verify that the test was successful by examining the output file. If the test is successful, continue to the next section.

Initialization: You must initialize the smart card or USB token first. This process is outside the scope of this paper. For information about how to enable new smart cards, see the smart card middleware documentation.

Configuring the certificate authority

Next, configure the certificate authority (CA) on the IMS Server:

1. Start the Microsoft Certification Authority by navigating to **Start** → **Administrative Tools** → **Certification Authority**, as shown in Figure C-2.

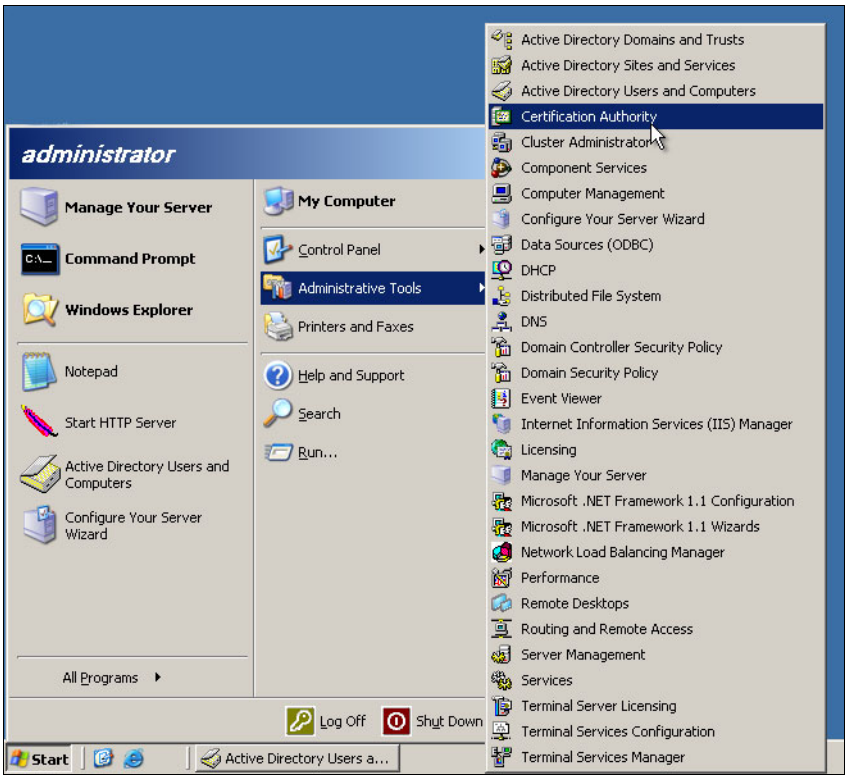

Figure C-2 Starting the Microsoft Certification Authority

2. A window opens that displays details about the CA. In the left pane, select the CA server, and then select the `Certificate Templates` directory. The available certificate templates are displayed in the right pane, as shown in Figure C-3.

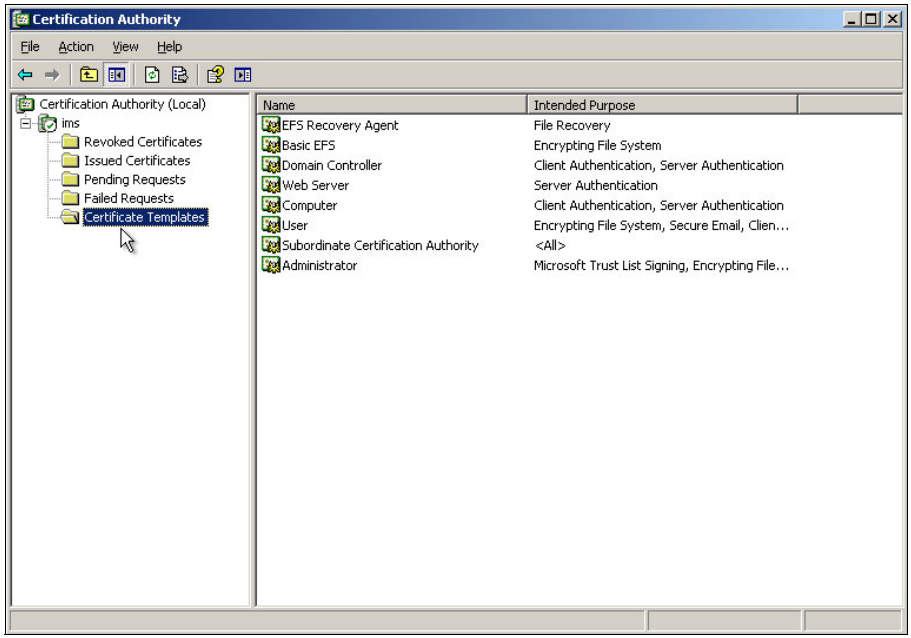

Figure C-3 Available certificate templates

3. To install the necessary templates (for example, the Smartcard User and Smartcard Logon templates), right-click in the right pane, and select **New** → **Certificate Template to Issue,** as shown in Figure C-4.

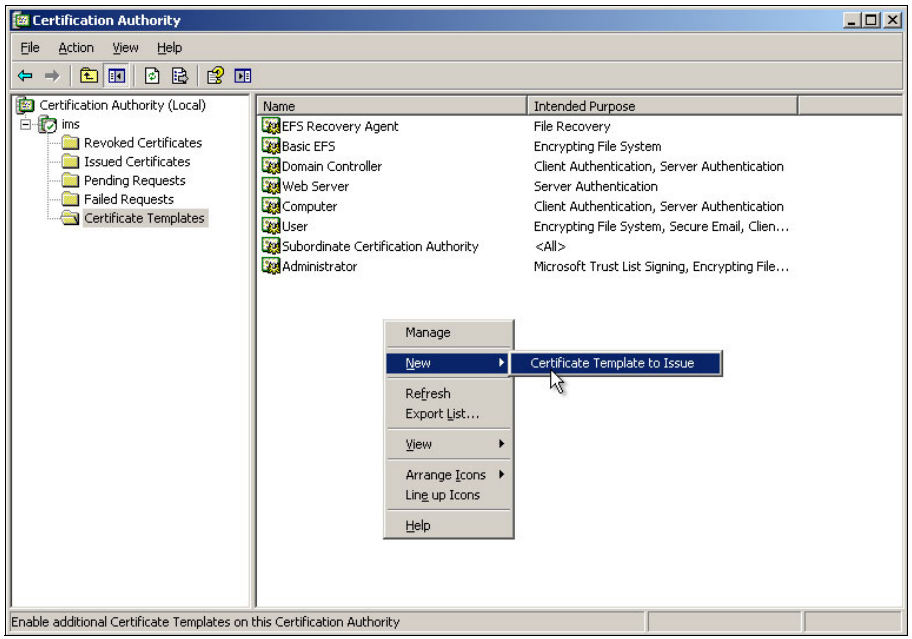

Figure C-4 New certificate

4. A list of available certificate templates appears. Scroll down, and select the **Smartcard Logon** and **Smartcard User** templates, as shown in Figure C-5. (You can select multiple certificate templates by pressing Ctrl.) Click **OK**.

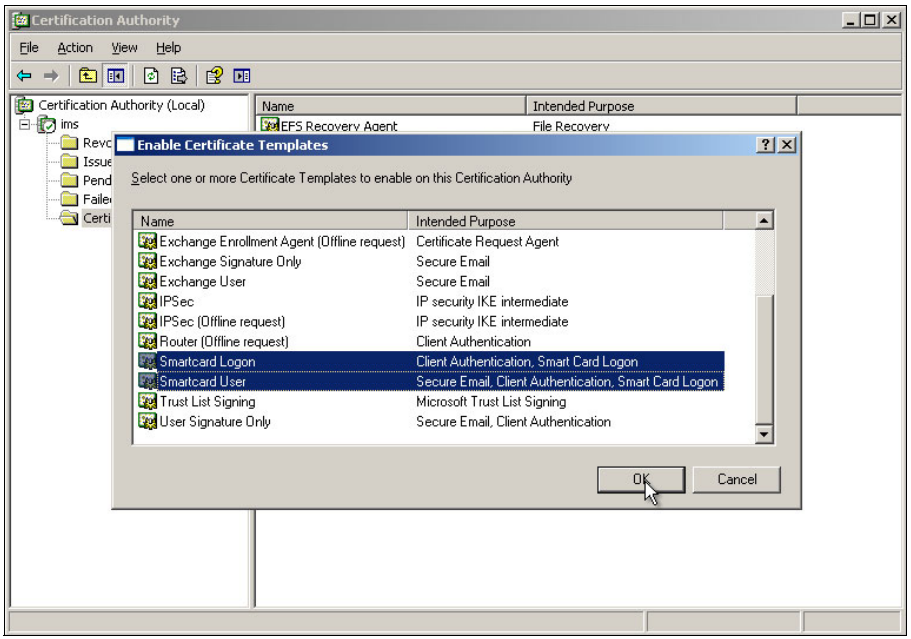

Figure C-5 Selecting the templates

The smart card templates are added to the Certificate Template list, and the server is ready to issue certificates, as shown in Figure C-6.

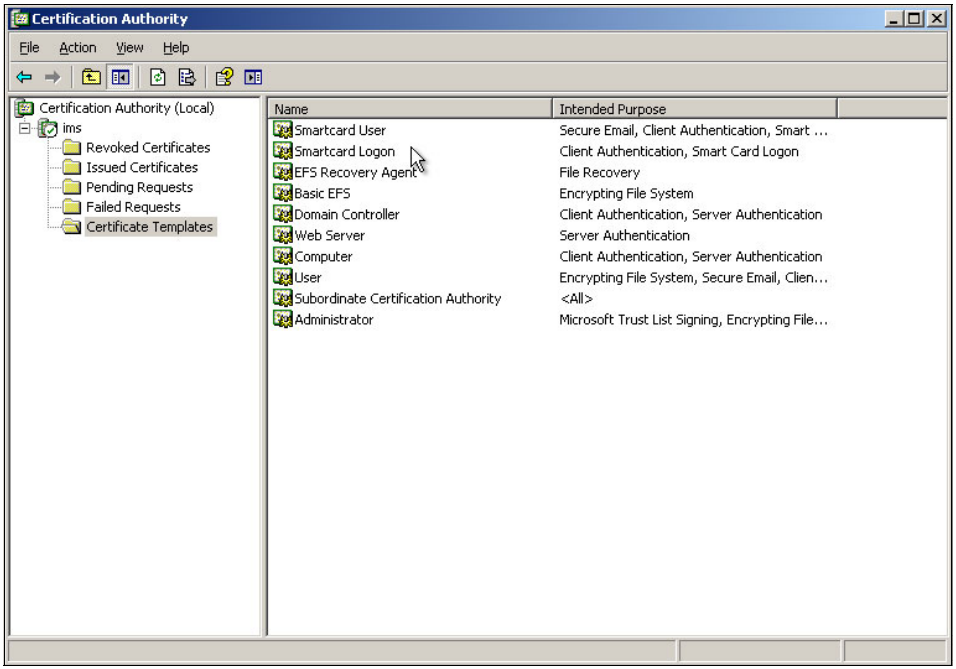

Figure C-6 Smart card certificates

Importing the CA root certificate to the HTTP Server truststore, part 1

Now, obtain the CA root certificate by clicking **Start** → **Administrative Tools** → **Active Directory Users and Computers**. The window shown in Figure C-7 on page 379 opens.

This scenario uses the Microsoft Certificate Server to obtain the domain name.

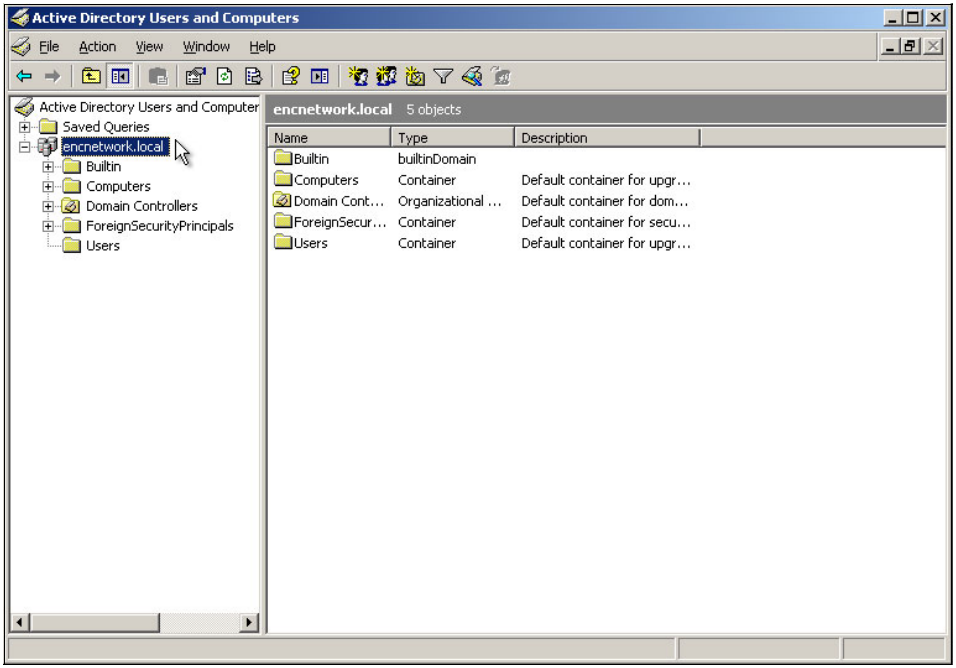

Figure C-7 Opening the Active Directory

Next, you need to obtain the Internet Information Services (IIS) server port number. By default, the IIS port number is 80. However, because the IBM HTTP Server already requires port 80, you need to modify the IIS port during installation.

To find the IIS server port number, follow these steps:

1. Go to **Start** → **Administrative Tools** → **Internet Information Services (IIS) Manager**, as shown in Figure C-8.

Figure C-8 Opening the IIS Manager

2. Click the plus sign (+) for the server from the left pane.
3. Open the **Web Sites** directory, and right-click **Default Web Site**. If there is more than one website, right-click the one that is available, not the one that is stopped.
4. Select **Properties**. The window shown in Figure C-9 opens. Several parameters appear, one of which is the TCP port. Note the value of the TCP port. (If the value is 80, change it to 81.)

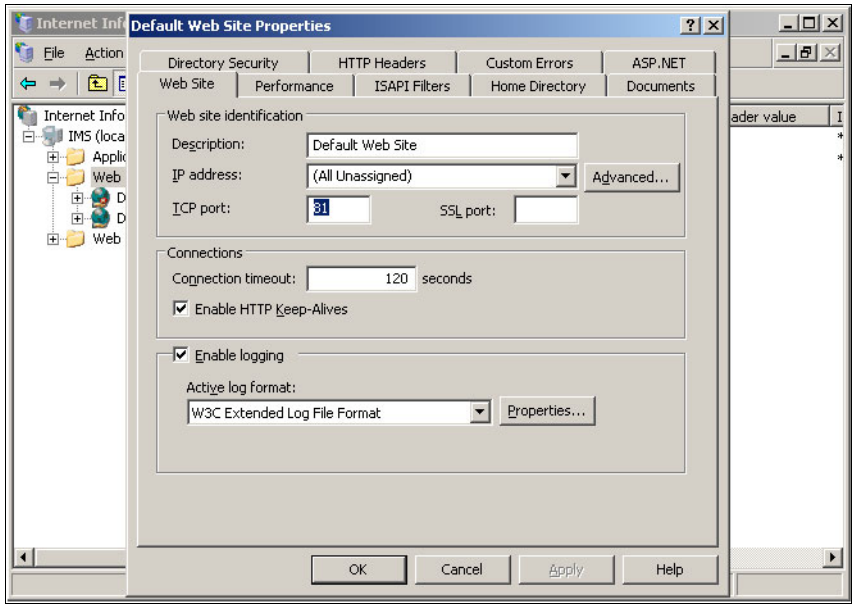

Figure C-9 TCP port value

Importing the CA root certificate to the HTTP Server truststore, part 2

After you determine the domain name and the IIS port number, enter the following address for the certificate server into the browser:

`http://`*`domain_name`*`:`*`IIS_port_number`*`/certsrv`

This address opens the CA server page and allows certificates to be issued.

Next, follow these steps:

1. Click the **Download a CA certificate, certificate chain, or CRL** link, as shown in Figure C-10.

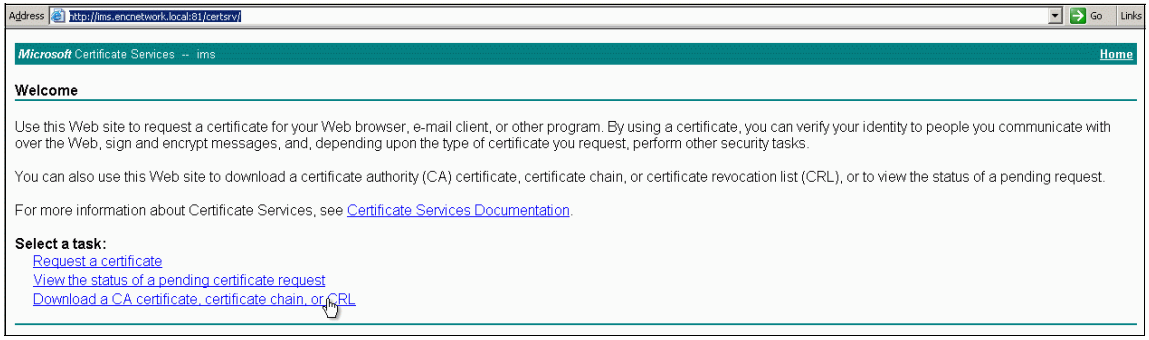

Figure C-10 Downloading a certificate

2. Enter the administrator user ID and password at the prompt.
3. At the next page, you are prompted to select an encoding method. The following standards are supported:
   - DER
   - Base 64

   This scenario uses the Base 64 standard. Select the **Base 64** option, and click **Download CA certificate**, as shown in Figure C-11 on page 383.

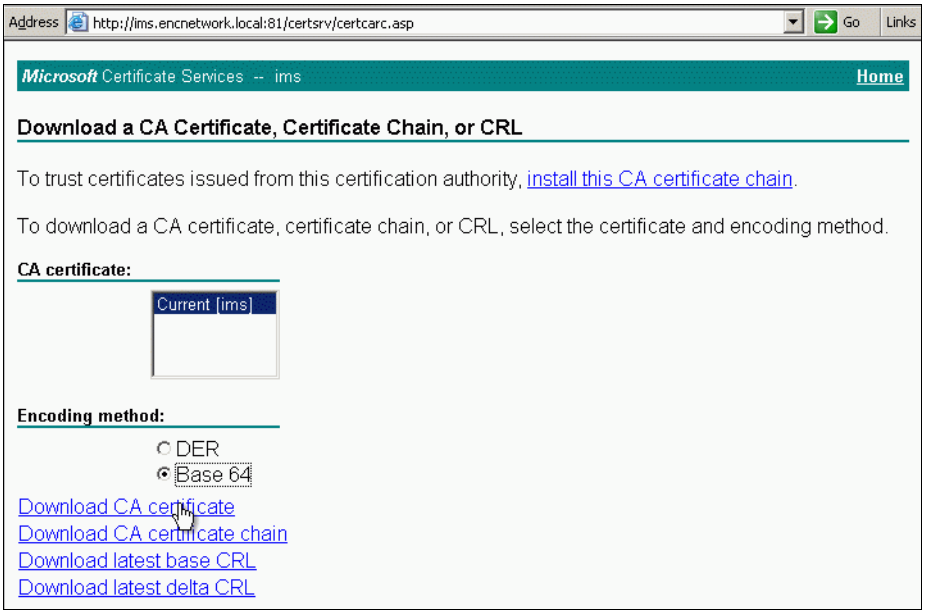

Figure C-11 Downloading the Base 64 CA certificate

4. In the confirmation box, click **Save**, and select the location where you want to save the certificate. Assign a name to the certificate, as shown in Figure C-12.

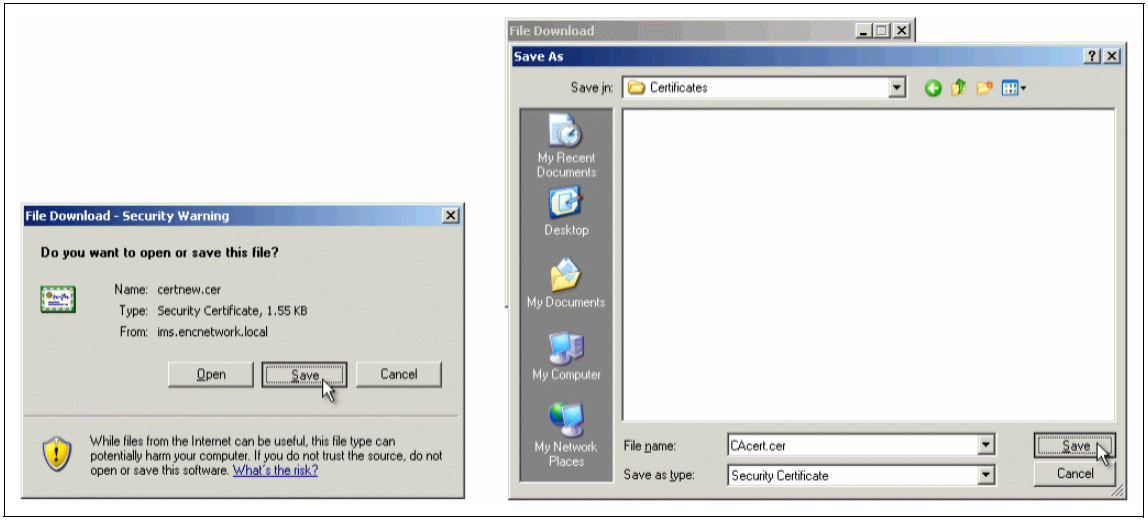

Figure C-12 Saving the certificate

5. After you obtain the root CA certificate, import it into the IBM HTTP Server truststore:

   Navigate to **Start** → **IBM WebSphere** → **Application Server v7.0** → **Profiles** → **AppSrv01** → **Administrative Console**.

   On the left pane, expand **Servers** and then expand **Server Types**. On the right pane, under Web servers, select the server that you want, as shown in Figure C-13 on page 385.

   WebSphere Application Server administrator credentials: At the Administrative Console, you need to enter the WebSphere Application Server administrator credentials.

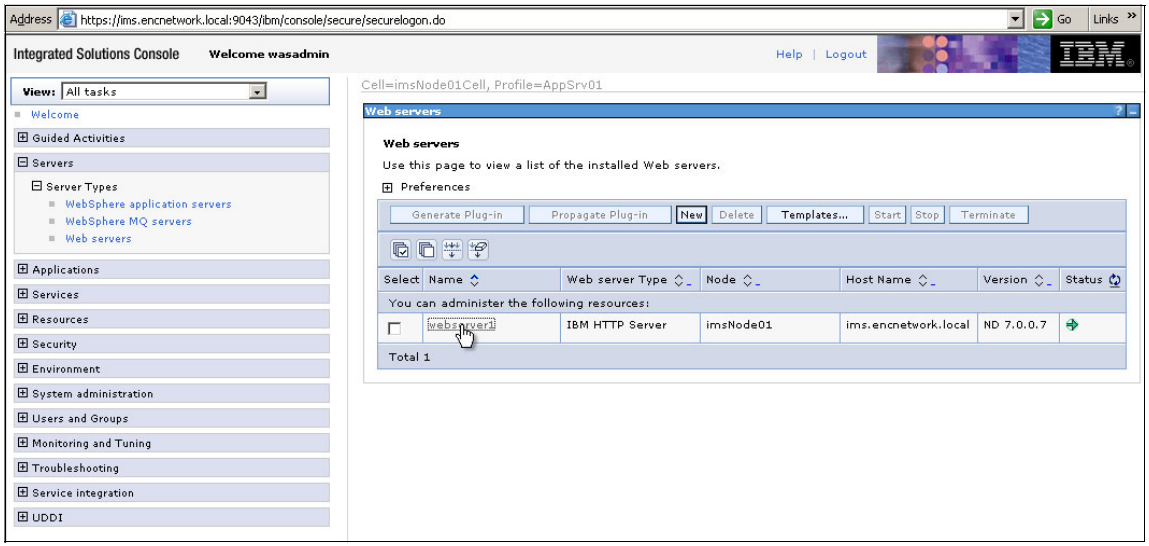

Figure C-13 Administrative Console

6. Under the Configuration tab, in the Additional Properties section, click **Plug-in properties**, as shown in Figure C-14.

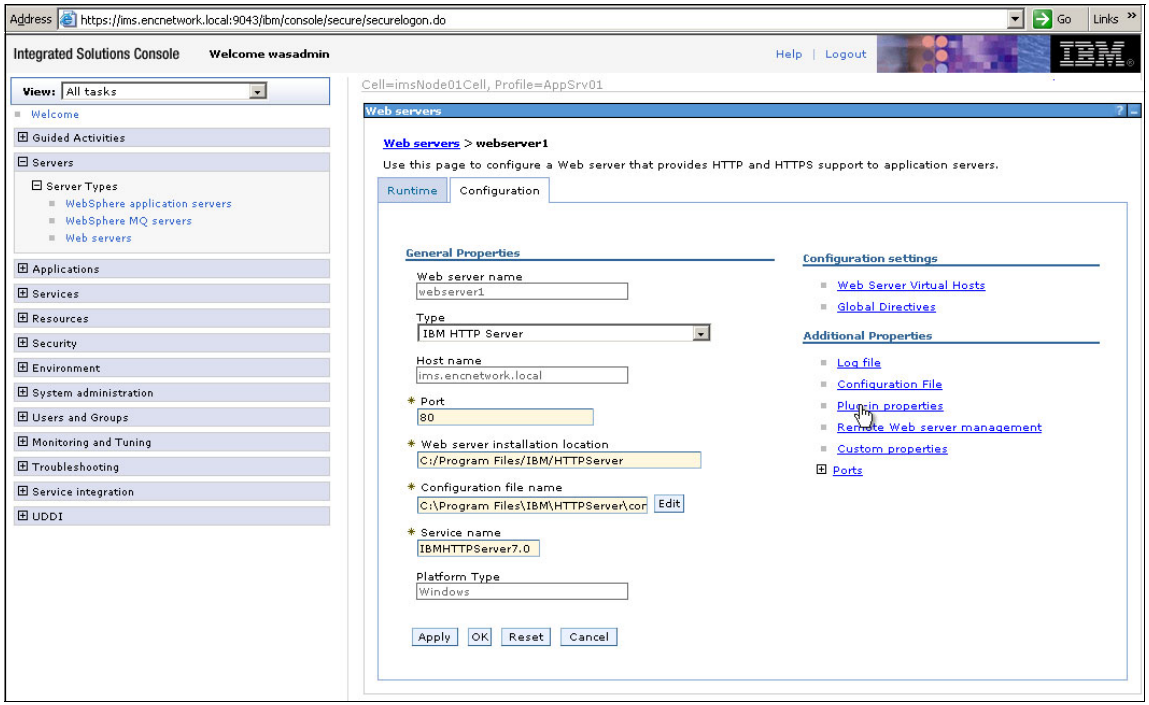

Figure C-14 Selecting Plug-in properties

7. Click **Manage keys and certificates**, as shown in Figure C-15.

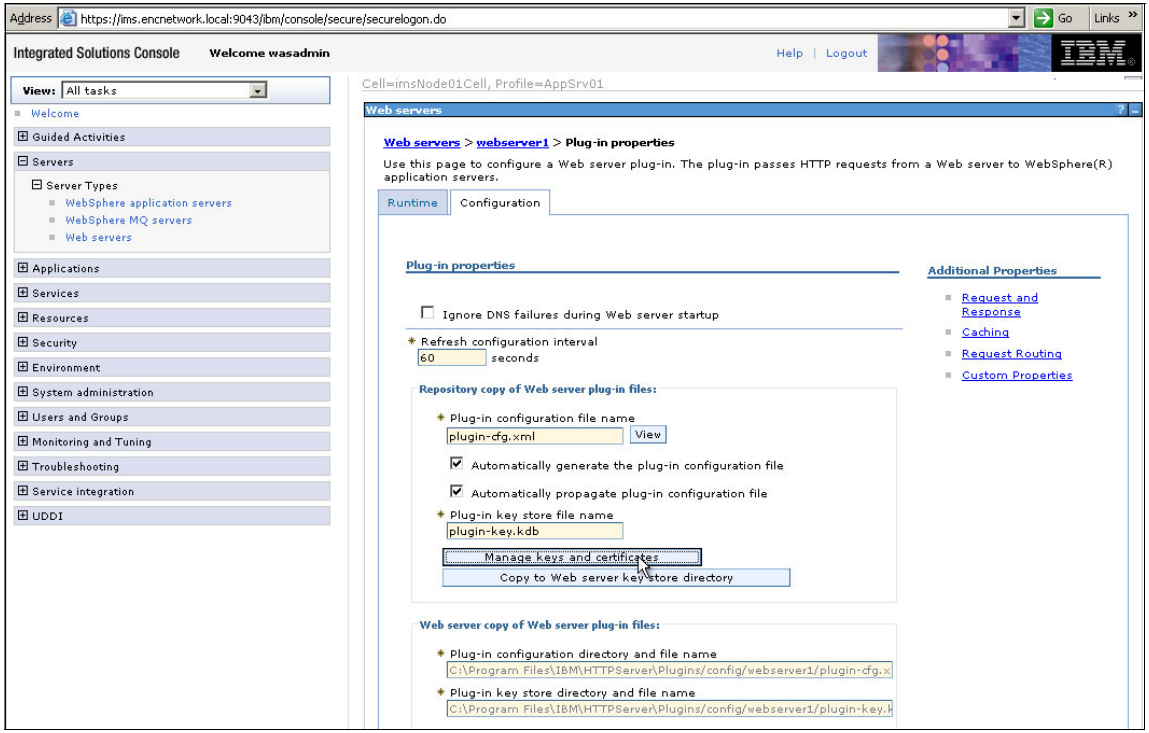

Figure C-15 Selecting Manage keys and certificates

8. Then, click **Signer certificates**, as shown in Figure C-16.

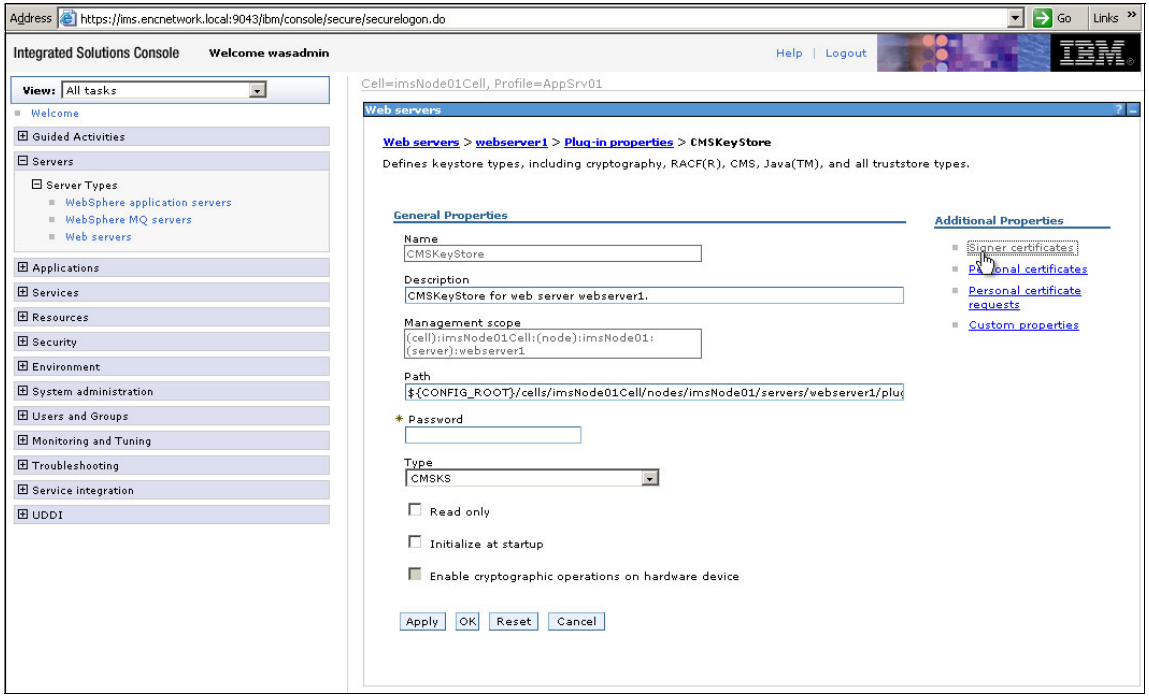

Figure C-16 Selecting signer certificates

9. On the right pane of the next panel, complete the following information:
   - **Alias**: An alias name of your choice
   - **File Name**: Full path of the CA certificate that you created earlier
10. Click **OK**, as shown in Figure C-17 on page 389. Save the changes when prompted to do so.

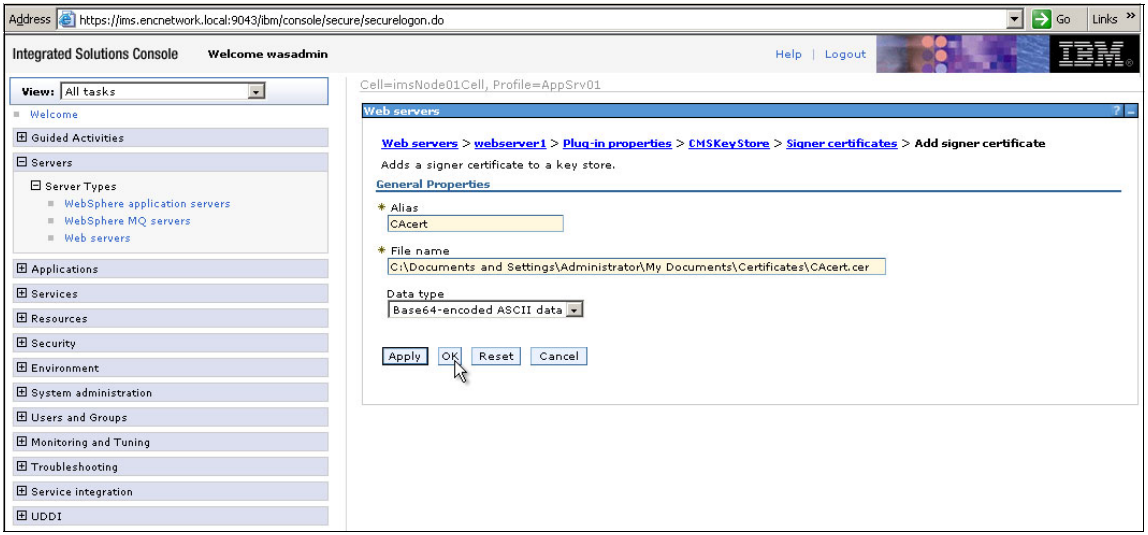

Figure C-17 Copying certificate to web server

11. On the left pane of the WebSphere Application Server Administrative Console, expand **Servers** and then expand **Server Types**. Select **Web servers**.

    On the right pane, select the web server that you want, and then, under Plug-in Properties, click **Copy to web server key store directory**, as shown in Figure C-18 on page 390. The root CA certificate is imported into the IBM HTTP Server truststore.

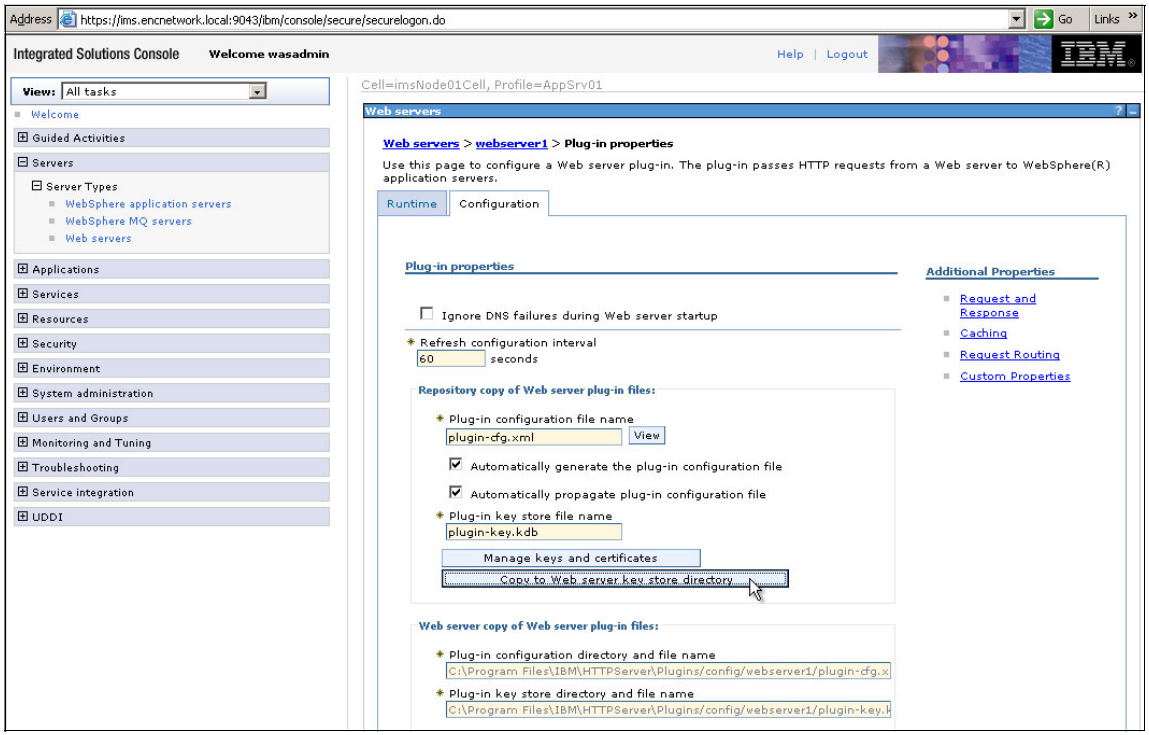

Figure C-18 Copying to keystore

12. Finally, restart the IBM HTTP Server. On the left pane of the WebSphere Application Server Administrative Console, expand **Servers** and then expand **Server Types**. Select **Web servers**.

    On the right pane, select the **webserver1** link, and click **Stop**. After webserver1 stops, select the link again, and click **Start**.

Enabling two-way Secure Sockets Layer on the IBM HTTP Server

To enable secure communications on the IBM HTTP Server, follow these steps:

1. Log on to the WebSphere Application Server Administrative Console.

2. Then, click **Server** → **Server Types** → **Web servers** → ***web_server***. Under Additional Properties, click **Configuration file**, as shown in Figure C-19.

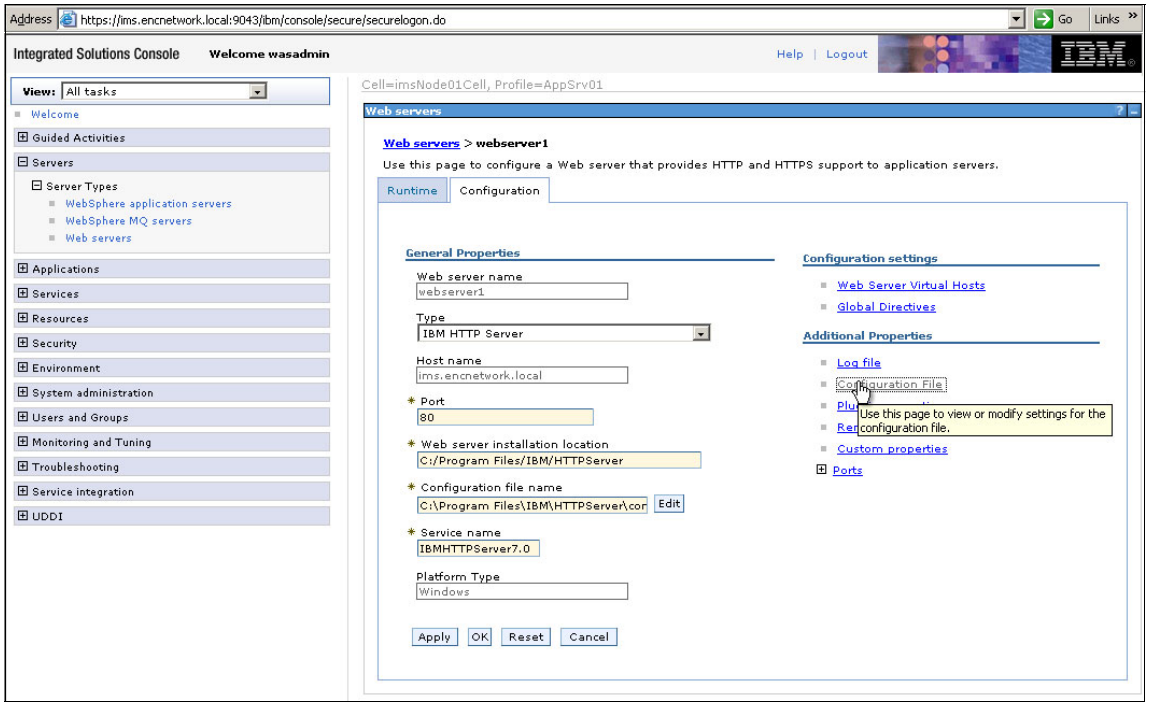

Figure C-19 Configuration file

3. Insert the following text between `SSLProtocolDisable SSLv2` and `SSLServerCert default`, as shown in Figure C-20:

   `SSLClientAuth optional`

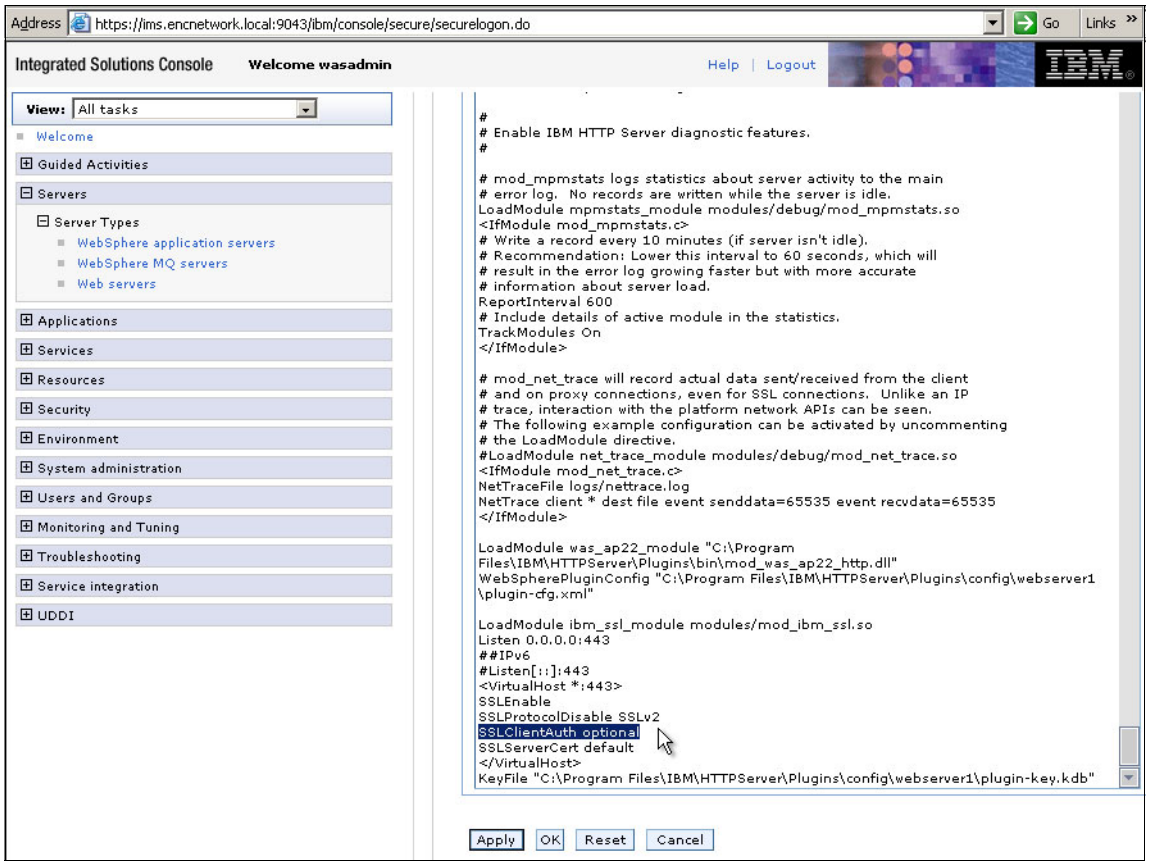

Figure C-20 Entering text

4. Click **OK**. Then, on the next page, click **OK** again, and click **Save**.

Creating IMS Server policies for smart card use

To create the IMS Server policies for smart card use, follow these steps:

1. Open a browser, and enter the IMS Server location. On the IMS Server page, select **AccessAdmin**.
2. Enter the login details for the administrator for the IMS Server.

3. On the left pane of the AccessAdmin panel, under Machine Policy Templates, click **New Template**.
4. On the Create new machine policy template panel, shown in Figure C-21, enter the following information:
   - Name: Name of the template. Assign a meaningful name.
   - Criteria: Indication that this template is for specific machines on your domain. Use the default option, which is **Use only machines that match these criteria**.
   - Authentication Policies: Enter `Smart card designation` in the text box.
5. Click **Add**.

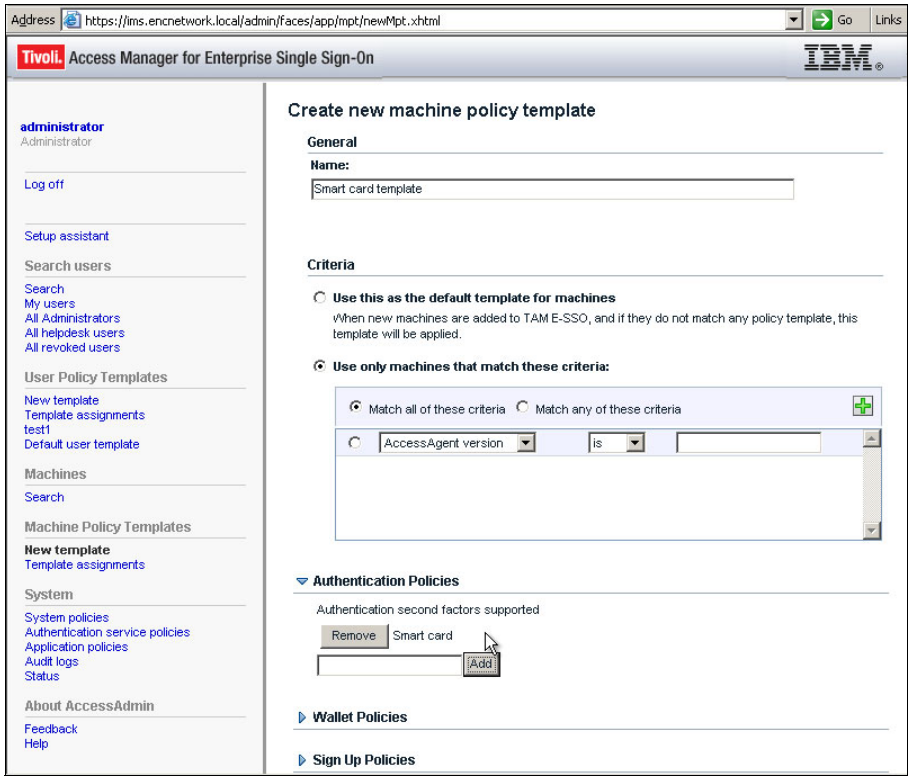

Figure C-21 Entering the new template information

6. Next, scroll further down on the Create new policy template panel, expand **AccessAgent Policie**s, and click **Smart card policies**.

7. When prompted, select **Yes** to enable Windows smart card logon, and then click **Add**, as shown in Figure C-22.

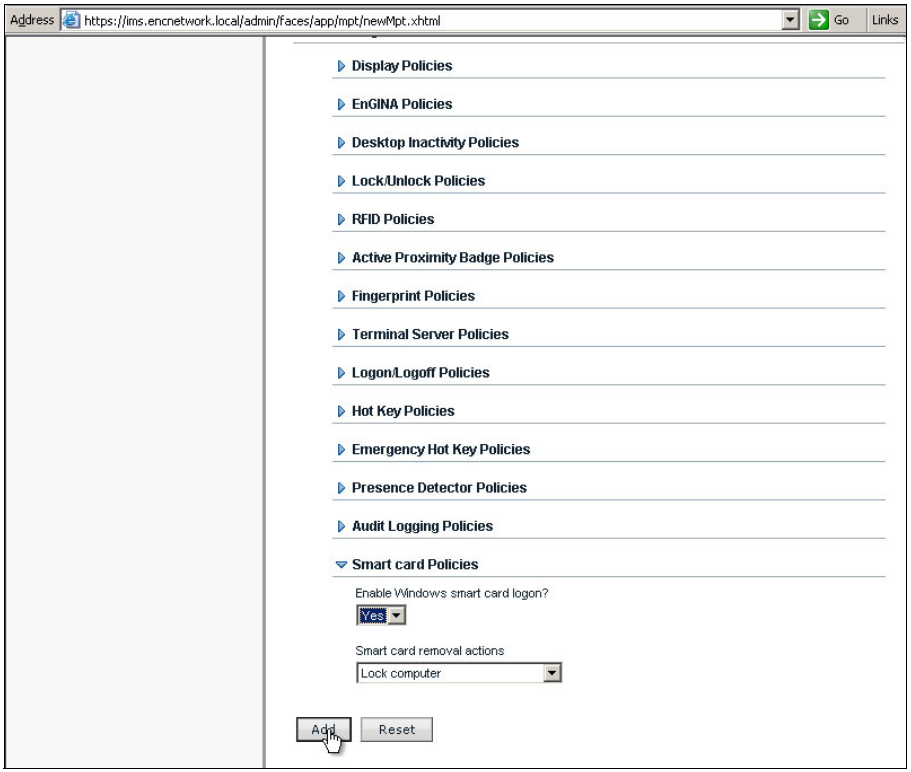

Figure C-22 Adding smart card policies

Assigning the new template to the client workstation

To assign the new template to the client workstation, go to **Machines** → **Search**. Enter an asterisk (*) in the Search for field and select **Host name** in the Search by drop-down list. Ensure that you select **All templates** in the Search in template drop-down list. Then, click **Search** to list the workstations that connect to the IMS Server by using AccessAgent, as shown in Figure C-23.

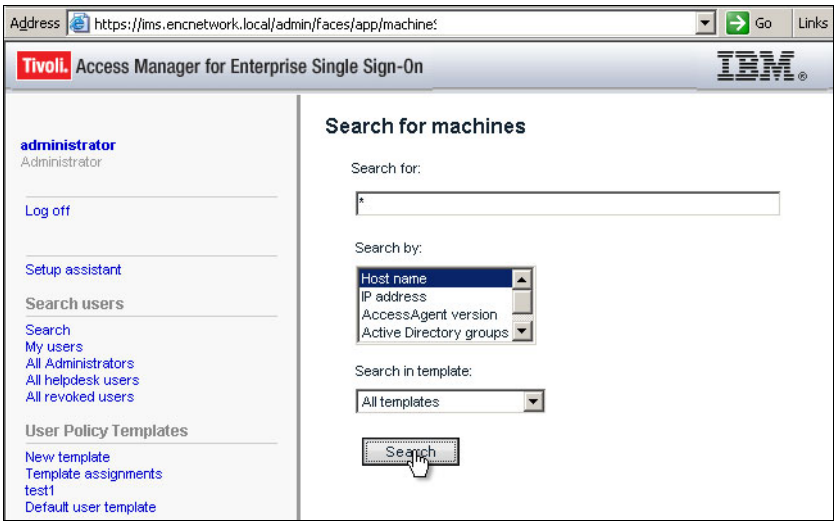

Figure C-23 Listing the workstations connected to the IMS Server by using AccessAgent

Select the workstation that you want. Then, from Machine Template Assignment, select **Smart Card policy**, and click **Assign**.

Modifying the user default template to accept smart card authentication

To modify the user default template to accept smart cards for authentication, follow these steps:

1. Under the Apply user policy templates heading, select **Default user template**.
2. On the new panel, click **Authentication Policies**. Then, enable **Smart card box**, and click **Update**.
3. Under the Search users heading, click the **Search** link. Then, type the required fields to refine your search and click **Search**.

4. Select the users who require smart card use. Under the Apply user policy template heading, select **Default user template** from the drop-down list, and click **Apply to selected results**.
5. At the confirmation prompt, click **OK**, as shown in Figure C-24.

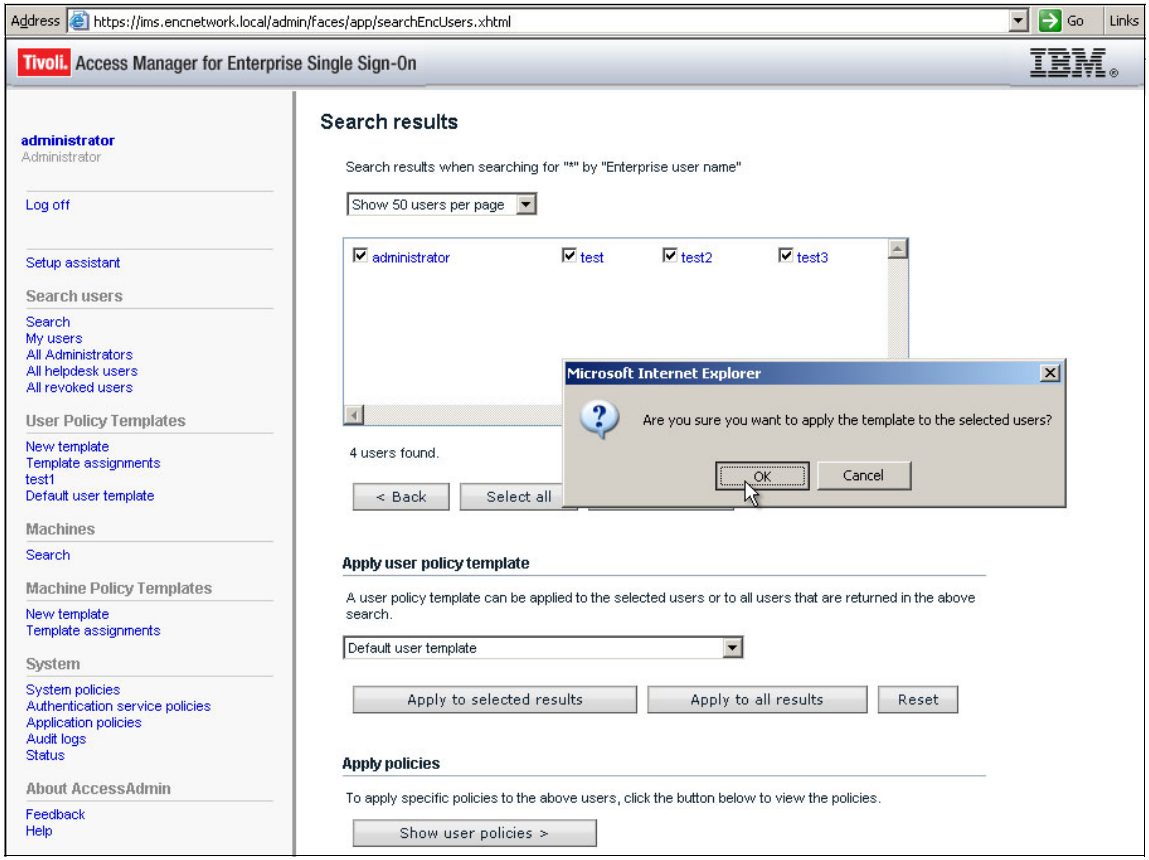

Figure C-24 Confirming modification to template

6. A status bar displays the progress of applying the user template. When the task is complete, restart WebSphere Application Server, as shown in Figure C-25.

   To stop WebSphere, select **Start** → **All Programs** → **IBM WebSphere** → **Application Server Network Deployment V7.0** → **Profiles** → **AppSrv01** → **Stop the server**.

   To restart WebSphere, select **Start** → **All Programs** → **IBM WebSphere** → **Application Server Network Deployment V7.0** → **Profiles** → **AppSrv01** → **Start the server**.

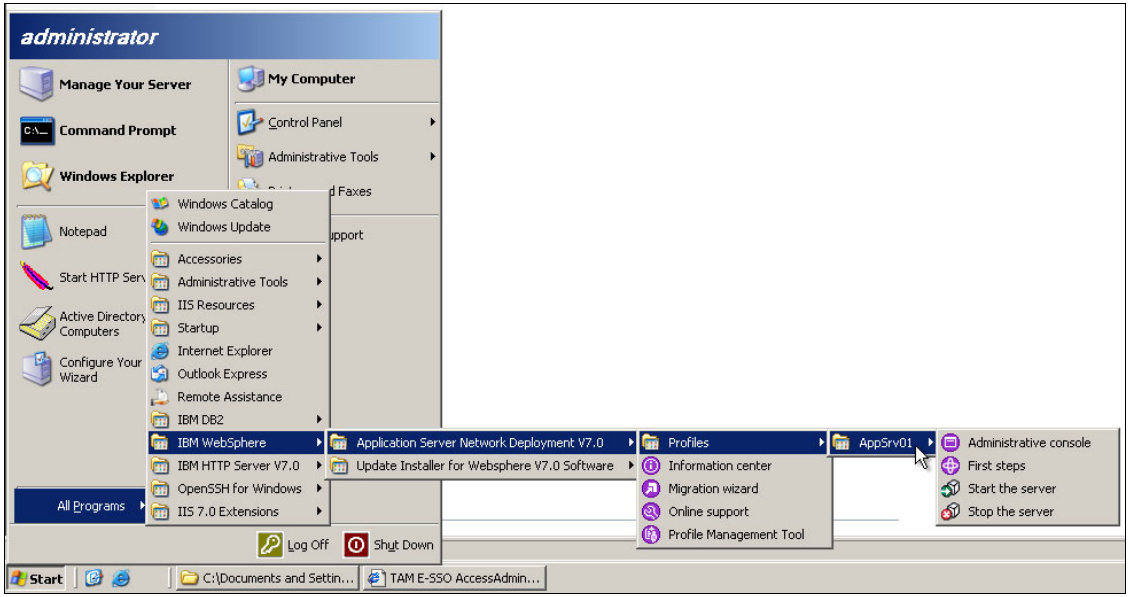

Figure C-25 Restarting WebSphere Application Server

7. The AccessAgent icon on the client system displays a message for the computer to be restarted due to changes on the IMS Server. Restart the computer. AccessAgent is now ready to allow authentication by smart cards.

Issuing a certificate to a smart card

This exercise issues a smart card to the user who logs on, which might not be the case in an actual client scenario. To issue a certificate to a smart card, follow these steps:

1. On the client system, log on to the Windows system with the ID of the user who requires the smart card use. Do not use AccessAgent to log in.

2. Insert the smart card in the reader or the token in a spare USB slot, as appropriate.
3. Go to the certificate server web page:

   `http://`*`domain_name`*`:`*`IIS_port_number`*`/certsrv`
4. Log on by using the user credentials, as shown in Figure C-26.

Figure C-26 Logging in to issue a certificate

5. Click **Request a certificate** from the Select a task options. Then, select **Advanced Certificate Request** and select **Create and submit a request to this CA**.

6. Change only the following parameters, as shown in Figure C-27:
   - For Certificate Template, select **Smartcard User**.
   - From the CSP drop-down list under Key Options, select the relevant middleware that is used within your environment. (This example uses the Charismathics Smart Security Interface CSP.)

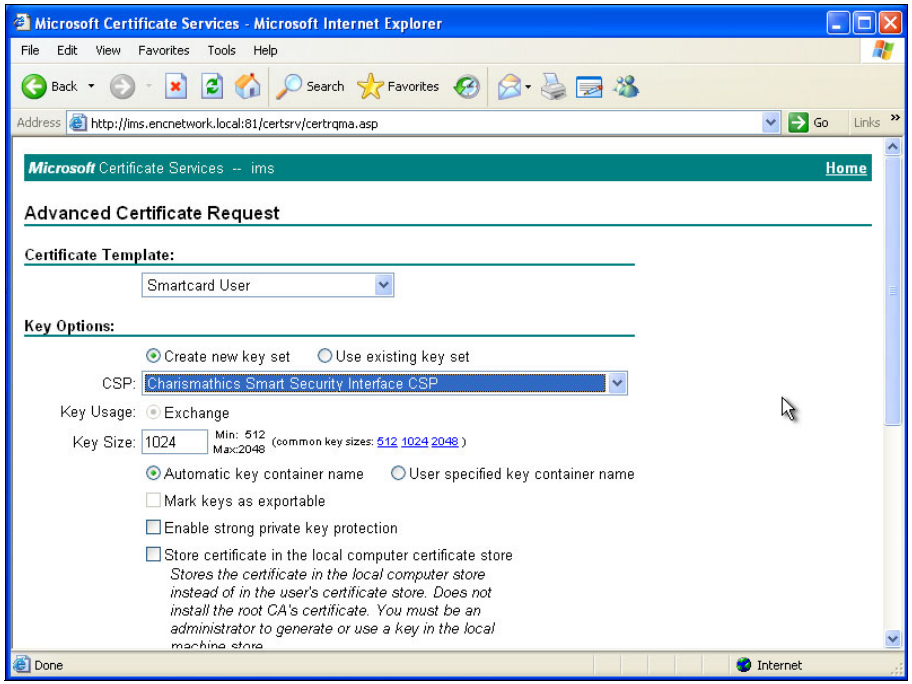

Figure C-27 Changing parameters to request a certificate

7. For Request Format under Additional Options, select **PKCS10** (Public Key Cryptography Standard for requesting certificates).
8. Click **Submit**, as shown in Figure C-28.

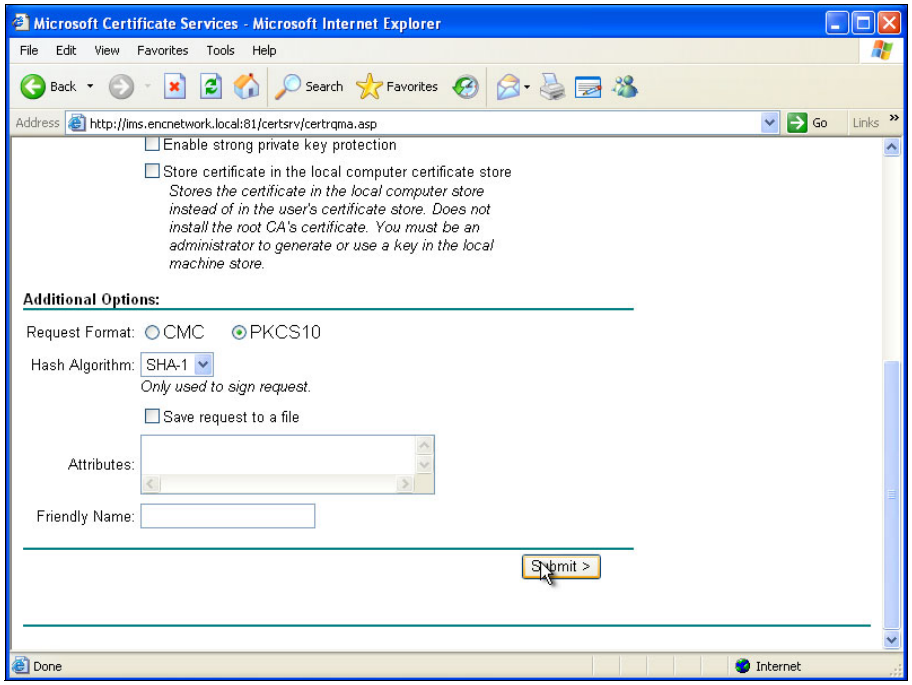

Figure C-28 Submitting certificate request

9. If you receive a warning message about a potential scripting violation, click **Yes** to open a window for the appropriate smart card middleware CSP. Then, enter the PIN for the smart card, and click **Login**, as shown in Figure C-29.

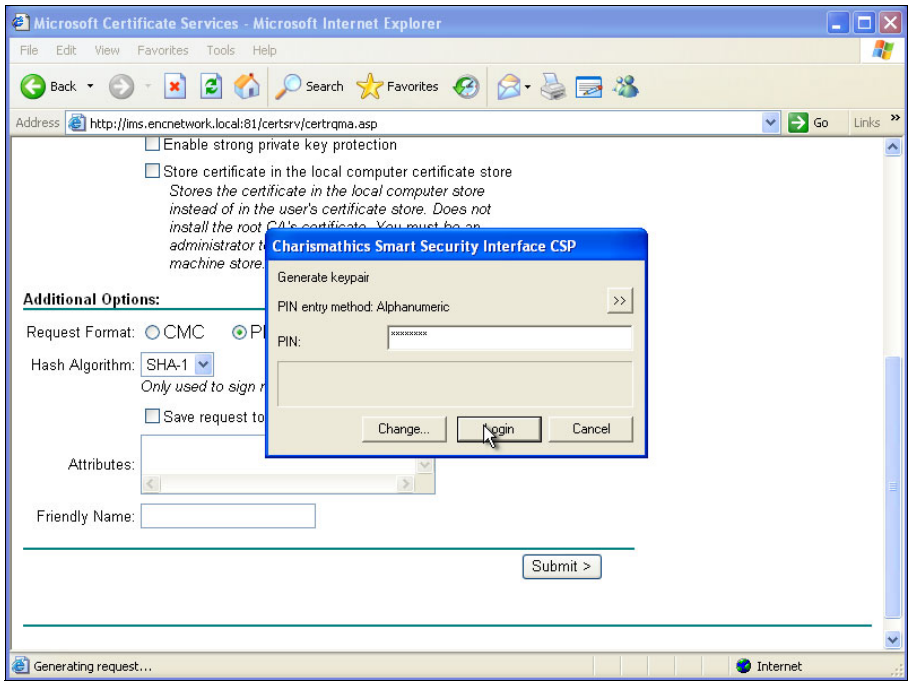

Figure C-29 Providing login information for the smart card middleware

The following message appears:

```
Generating Request
```

Then, another message appears:

```
Waiting for Server response
```

These messages might display for 2 minutes or so.

10. When the Certificate Issued page appears, click **Install this certificate**. When a warning message about a potential scripting violation appears, click **Yes** to continue. Then, when a message appears requesting confirmation for the installation of the certificate from the CA server, click **Yes** again.

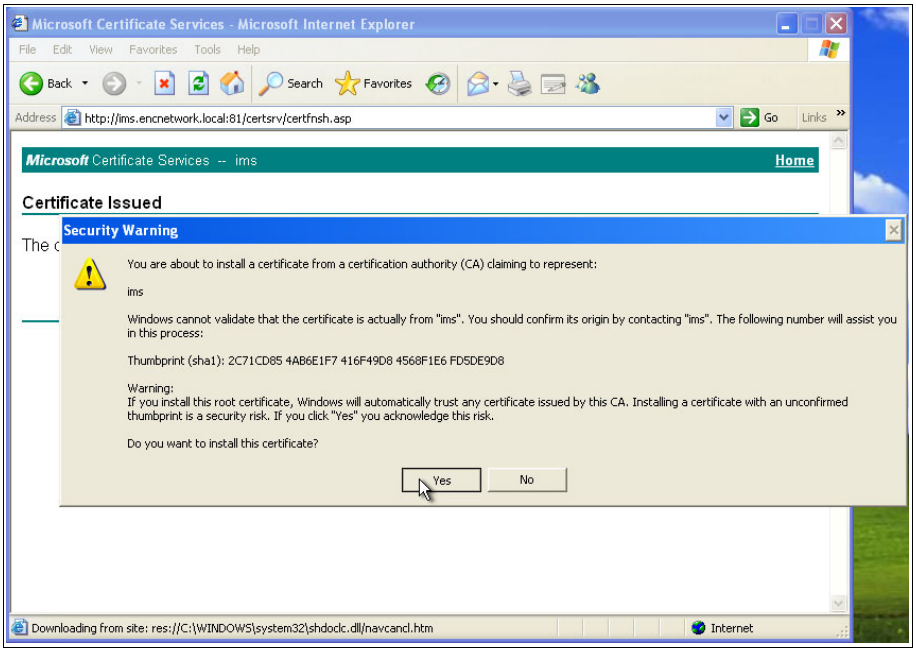

Figure C-30 Accepting a new certificate from your IMS Server

11. Click **Yes** when prompted to save the CA certificate.

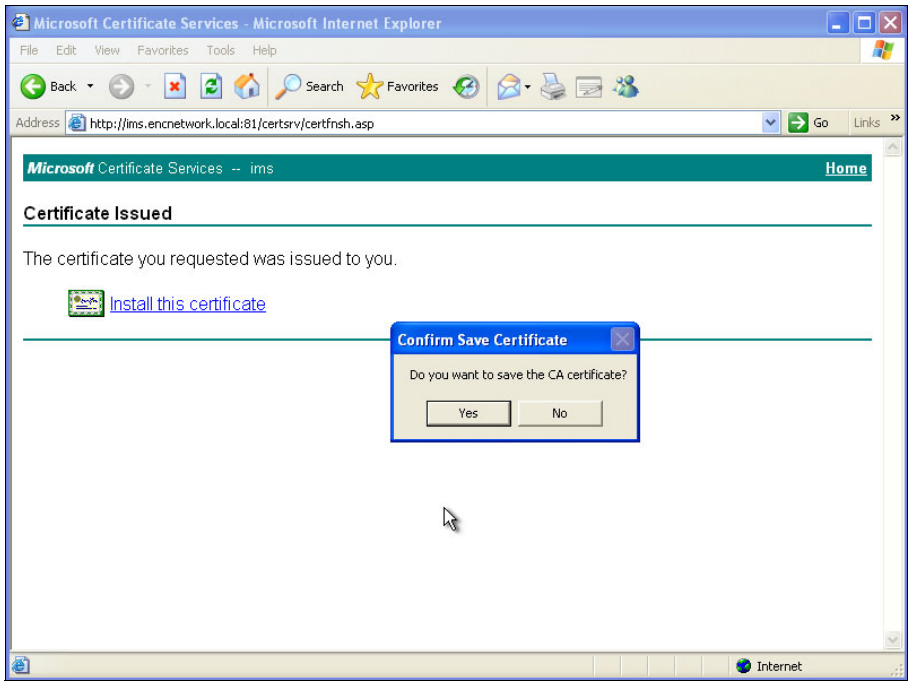

Figure C-31 Saving the CA certificate

12. After the confirmation message appears, open the middleware software and read the smart card contents. Verify that the certificate is installed. Figure C-32 shows an example of the contents as displayed by the Charismathics Smart Security Interface Manager.

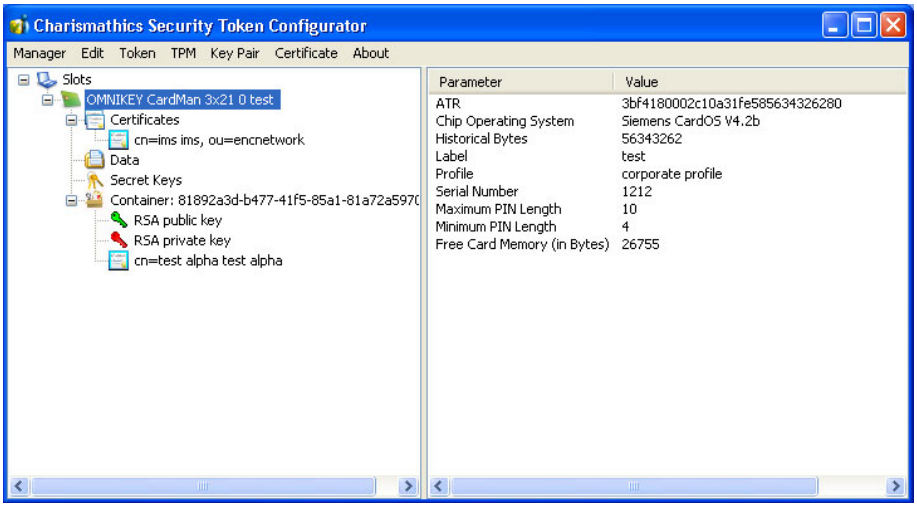

Figure C-32 Contents displayed by the Charismathics Security Token Configurator

Registering a smart card to a user

To register a smart card to a user, follow these steps:

1. On the client system, insert the smart card into the reader when the AccessAgent logon window opens. AccessAgent prompts you for the smart card registration with the user account, as shown in Figure C-33.

Figure C-33 Smart card registration

2. Enter the PIN that was assigned to the smart card during the certificate request process, and then click **OK**, as shown in Figure C-34.

Figure C-34 Entering smart card PIN

3. A prompt to register the smart card with the IMS Server appears. You can use the smart card with AccessAgent to log on, as shown in Figure C-35. Then, click **Next**.

Figure C-35 Logging on with AccessAgent

4. When prompted, click **Yes** if you registered the user earlier to use IBM Security Access Manager for Enterprise Single Sign-On, as shown in Figure C-36. (Otherwise, click **No** and AccessAgent enrolls the user with the IMS Server.)

Figure C-36 Registration to use IBM Security Access Manager for Enterprise Single Sign-On

5. Enter the account details, as shown in Figure C-37, and click **OK**.

Figure C-37 Entering account details

The credentials are injected automatically into the Windows system logon prompt, as shown in Figure C-38.

Figure C-38 Windows system logon credentials

You are now logged on, and your registration for smart card use is complete. If you remove the smart card, you are required to present the smart card and to enter the PIN the next time that you log on with this account.

Configuring authentication to use radio frequency identification cards

This section explains how to configure an existing IBM Security Access Manager for Enterprise Single Sign-On environment to use radio frequency identification (RFID) cards as additional authentication factors. It includes the following topics:

- "Prerequisite environment" on page 408
- "Creating and assigning the RFID machine policy template" on page 409
- "Creating an authentication code for the user" on page 412
- "Registering the RFID card to the user" on page 414

Prerequisite environment

For platform requirements and configuration instructions, see the IBM Security Access Manager for Enterprise Single Sign-On product documentation.

To perform this exercise, you need the following resources:

- Integrated Management System Server (IMS Server):
  - Microsoft Certificate Server
  - Internet Information Services
  - IBM Security Access Manager for Enterprise Single Sign-On IMS Server prerequisites:
    - WebSphere Application Server
    - IBM HTTP Server
    - A supported database (for example, DB2)
  - Smart card middleware
- Client:
  - IBM Security Access Manager for Enterprise Single Sign-On AccessAgent
  - Smart card middleware
  - Initialized smart card and reader or USB token
  - Drivers for reader or token
  - RFID card with reader
  - Drivers for RFID reader

Creating and assigning the RFID machine policy template

To create and assign the RFID machine policy template, follow these steps:

1. Open a browser and enter the IMS Server location. From the IMS Server page, click **AccessAdmin**. Then, enter the login details for the Administrator for the IMS Server.
2. On the AccessAdmin page, under Machine Policy Templates, click **New Template**. Then, enter the following information, as shown in Figure C-39 on page 410:

   Name: Enter the name of the template. Assign a meaningful name.

   Criteria: Specify the criteria if this template is for specific machines on the domain. Use the default option, which is **Use only machines that match these criteria**.

   Authentication Policies: Select **RFID**.

3. Click **Add**.

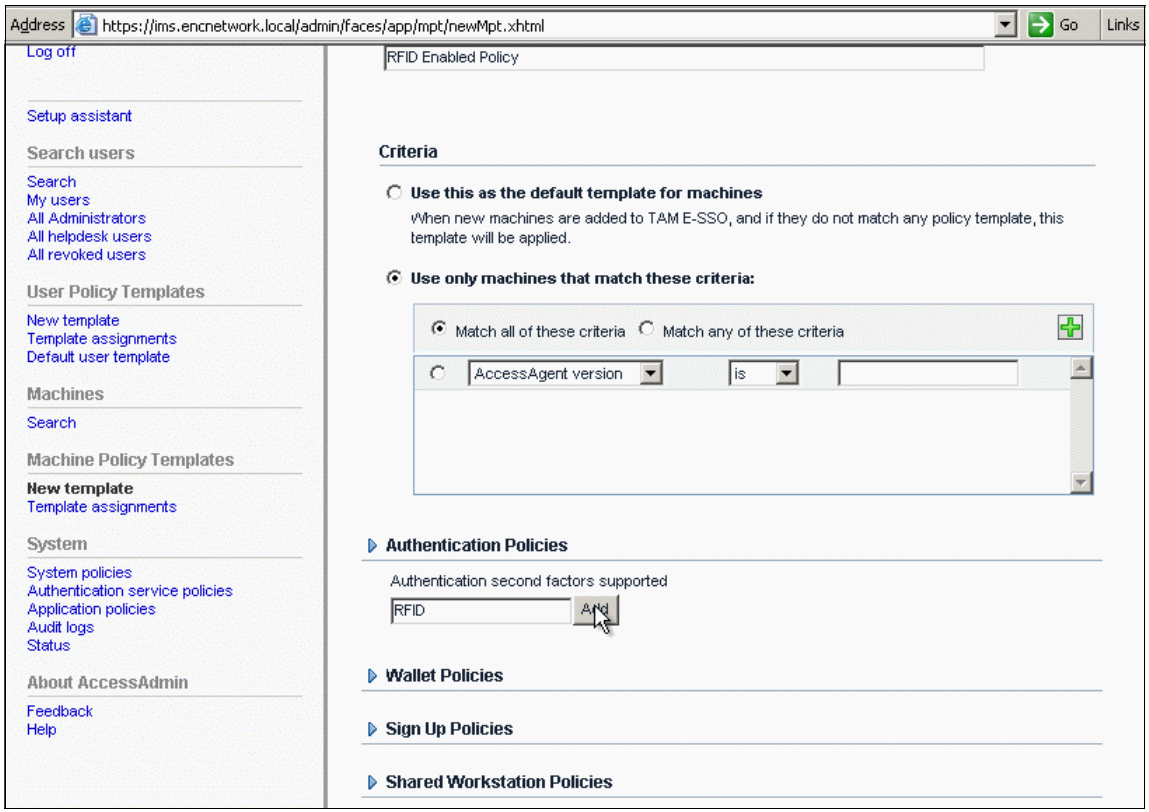

Figure C-39 Entering the new template information

4. Add the policy template by again clicking **Add**, as shown in Figure C-40.

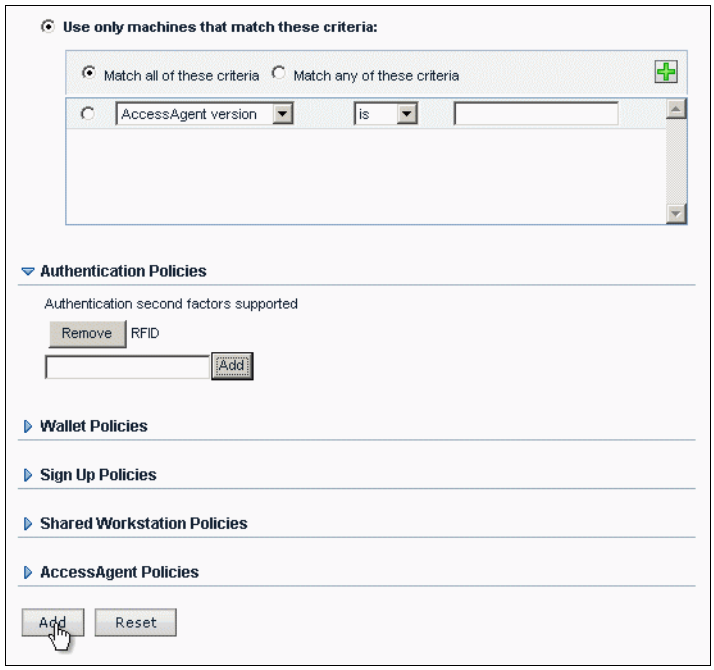

Figure C-40 Adding the template

5. Now, assign this new template to the client system to use when the user with the RFID badge logs on. Complete these tasks:
   - Go to **Machines** → **Search** on the left pane.
   - Enter an asterisk (*) in the Search for field.
   - Select **Host name** in the Search by drop-down list.
   - Ensure that you select **All templates** in the Search in template drop-down list.
   - Then, click **Search** to list the workstations that connect to the IMS Server by using AccessAgent, as shown in Figure C-41 on page 412.

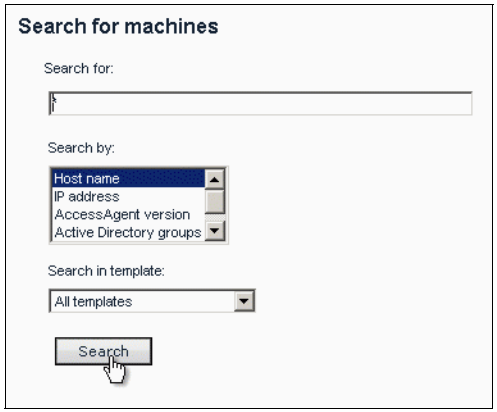

Figure C-41 Listing the workstations

6. Select the workstation. Then, from the Machine Template Assignment list, select the RFID policy, and click **Assign**.

Creating an authentication code for the user

Follow these steps to create an authentication code to permit a user to log on with an RFID badge as an authentication factor:

1. In AccessAdmin, select **IMS Server** → **AccessAdmin**. Locate the user for whom you want to create an authentication code (here, `A B`), as depicted in Figure C-42 on page 413.

Figure C-42 Logging on to AccessAgent

2. Scroll down to Helpdesk Authorization and select **Issue authorization code**. Note the authorization code that is generated (Figure C-43).

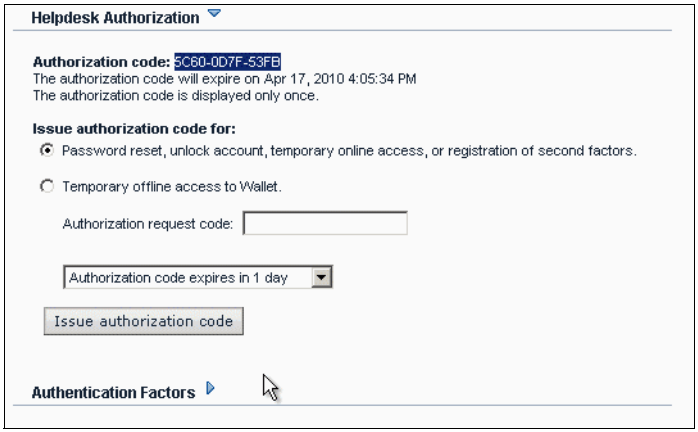

Figure C-43 Issue an authorization code

Registering the RFID card to the user

To register the RFID card to the user, follow these steps:

1. On the client system, tap your RFID card on the reader when prompted at the AccessAgent logon window. AccessAgent prompts you for the RFID card registration with the user account, as shown in Figure C-44.

Figure C-44 Prompt to tap RFID card

2. When prompted, select **Yes** if the user is already registered with IBM Security Access Manager for Enterprise Single Sign-On. (Otherwise, select **No**, and AccessAgent enrolls the user.) Then, click **Next**, as shown in Figure C-45.

Figure C-45 Register the RFID card

3. Enter the account details for User name, Password, and Domain. Then, click **OK**, as shown in Figure C-46.

Figure C-46 RFID card logon

4. As shown in Figure C-47, enter the authorization code that was generated previously (as described in “Creating an authentication code for the user” on page 412). The credentials are injected automatically into the Windows system logon.

Figure C-47 Entering authorization code

You are now logged on, and registration for RFID card use is complete. When you log on again with this account, you are required to present the RFID card and enter the password.

Strong authentication by using biometrics

This section includes the following areas:

- Finger biometrics-based authentication, which provides a brief overview of finger biometric-based authentication for IBM Security Access Manager for Enterprise Single Sign-On
- Configuring authentication to use fingerprint recognition, which provides detailed instructions for the installation and configuration of the required software components

Finger biometrics-based authentication

As depicted in Figure C-48 on page 417, finger biometrics-based authentication for IBM Security Access Manager for Enterprise Single Sign-On provides the same access to the Windows desktop and to enterprise applications as a single sign-on password, but without requiring keyboard entry of any password. With

this integrated solution enabled, the user is positively identified and authenticated at the Windows logon and at any point that authentication is required for sign-on to an application or database.

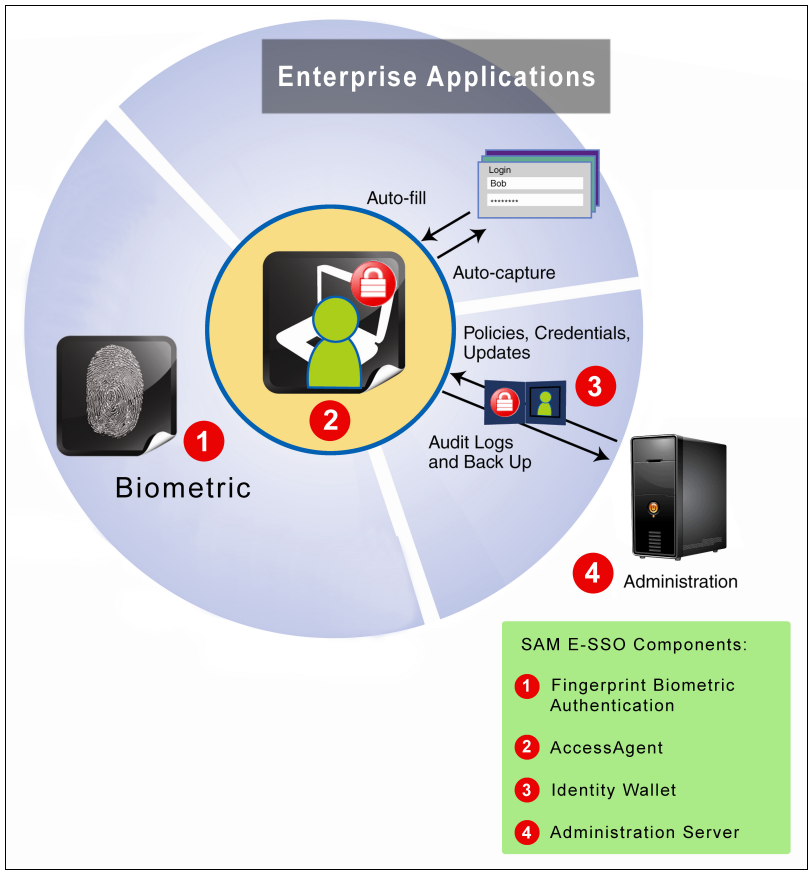

Figure C-48 Finger biometric authentication

Next, we describe BIO-key Biometric Service Provider from BIO-key International, Inc[1].

BIO-key Biometric Service Provider

BIO-key Biometric Service Provider is a set of software services that are integrated with the IBM Security Access Manager for Enterprise Single Sign-On user authentication process and the client and server software components that manage that process. These services were built with BIO-key's Biometric Service Provider (BSP) software development kit (SDK). BIO-key Biometric Service

[1] For more information about the BIO-key offerings, see this website: http://www.bio-key.com

Provider supports the Biometric Service Provider application programming interface (API) standard Version 1.1, as defined by the BioAPI Consortium[2].

How finger biometrics-based authentication works

Through an initial enrollment process, the employee's finger is scanned and digitized. Typically, a primary finger is designated, and a second finger is scanned for alternate use. The data extracted from the finger scan is converted into a mathematical model, which is used to build a registration template that represents the features, or minutiae, of the fingerprint.

When that user signs on to an IBM Security Access Manager for Enterprise Single Sign-On-enabled system or application, the finger is scanned and digitized again and a reference template is created. That template is then compared with the enrollment database to identify a positive match for user authentication.

Finger-based biometric authentication for IBM Security Access Manager for Enterprise Single Sign-On can be used on any private, shared, or roaming desktop. In addition to finger-based biometric authentication sign-on, you can use finger-based biometric authentication for quickly locking and unlocking the desktop. You can also use finger-based biometric authentication for switching between users, which is especially important in a shared workstation environment.

Components

BIO-key Biometric Service Provider consists of two primary services, *image capture* and *image matching*. After these services are installed and configured, they can be called only by IBM Security Access Manager for Enterprise Single Sign-On.

Image capture service

The BIO-key Biometric Service Provider image capture service manages the process of collecting the finger image from the user through the IBM Security Access Manager for Enterprise Single Sign-On Authentication Device Manager, digitizing that image, extracting key features, and creating a mathematical model or template. When installed and enabled, this service is embedded in the AccessAgent client.

The BIO-key patented image processing technology uses more than 40 levels of image enhancement, ridge, minutia, and vector data to extract useful data from the raw fingerprint, creating a highly discriminate template. This technology

[2] The BioAPI Consortium, of which BIO-key is a member, was formed to establish an industry standard API to interface with today's most common biometric technologies. For more information about the BioAPI Consortium, see its website: http://www.bioapi.org

virtually eliminates the possibility of either a "f*alse acceptance*" or "*false rejection*" response. A false acceptance means that an individual is incorrectly identified and authenticated as a user. A false rejection means that a valid user was incorrectly rejected because a match cannot be made.

BIO-key Biometric Service Provider currently supports more than 55 fingerprint readers from most major manufacturers, and most embedded readers shipped with notebooks and other workstations.

Image matching service

The BIO-key Biometric Service Provider image matching service manages the process of comparing the BIO-key highly discriminate templates by using the BIO-key patented algorithm. The BIO-key Biometric Service Provider image matching service compares over 1,500 points of data after the data is received from the image capture service to identify a matching fingerprint, if one exists. In a typical installation, the BIO-key Biometric Service Provider image matching service is embedded in both the IMS Server and AccessAgent on the client workstation.

BIO-key Biometric Service Provider process flow

The BIO-key Biometric Service Provider image capture and image matching services are invoked in the process flow at initial enrollment and subsequent sign-on.

User enrollment

The BIO-key Biometric Service Provider image capture service manages the process of capturing the finger image at initial enrollment (or whenever the user enrolls another fingerprint) and creating the registration template. To associate the registration template created at enrollment with the correct user, IBM Security Access Manager for Enterprise Single Sign-On uses the user account created during user registration. This user account can optionally be associated with an ActiveDirectory repository or other Lightweight Directory Access Protocol (LDAP)-based repository.

The windows for registering the user's fingerprint and associating it with the user's name and password are already built into AccessAgent. From the main Welcome window, the user scans the finger to be registered. AccessAgent then prompts the user for the user's IBM Security Access Manager for Enterprise Single Sign-On user name. After the user name is provided, AccessAgent verifies that the fingerprint is not already registered and prompts the user for the user's IBM Security Access Manager for Enterprise Single Sign-On password. After the user is authenticated, the user selects the appropriate finger (right index finger, for instance) from an image of two hands, as shown in Figure C-49.

Thereafter, the individual places the appropriate finger on the reader, and the information is instantly collected.

Figure C-49 Finger image capture

The number of fingers that can be captured is configurable from 1 to 10. As a preferred practice, a registration template of at least one finger from each hand is created so that a second fingerprint is available for match if the primary finger is bandaged or otherwise unusable for matching.

Quality control is built into the enrollment workflow to ensure that data is collected. The BIO-key Biometric Service Provider image capture service is pre-configured to scan the enrollee's finger three times and select the best capture for the registration template. AccessAgent prompts the user to scan the same finger a fourth time to verify the correctness of the captured registration template. If any of the scanned images are invalid, AccessAgent provides the appropriate error message (finger scanned too much to the right, for instance) and prompts for resubmission.

Enrollment can be completed on a dedicated workstation (for managed enrollment, for example) or on the user's workstation (for self-enrollment). If self-enrollment is used and finger image capture is successful, the user is logged in to AccessAgent for immediate access to authorized system resources, as evidenced by the system tray icon in the notification area of the Windows Desktop.

During enrollment, the workstation or other device must be connected to the network and to the IMS Server so that the registration template can be transmitted and stored with other templates of approved users in the IMS Server database.

As a configuration option, this registration template can also be cached in the user's AccessAgent Wallet on the workstation. Caching the registration template has two purposes:

- It enables login without keyboard entry of the user name.
- It might be used for faster user access to the desktop (or in other circumstances described next).

Subsequent to initial enrollment, if the user's privileges are revoked (for example, no longer employed or no longer granted access to system resources), the individual's registration templates are removed from the IMS Server database when the user is deleted (typically, by the administrator or the Help Desk). If the user re-enrolled on another workstation (creating another template), the template is replaced in the IMS Server database. Revoked and replaced registration templates are removed from the cached Wallet on the workstation whenever that user is connected to the IMS Server for authentication.

See "Configuring authentication to use fingerprint recognition" on page 425 for complete enrollment instructions, including the representative AccessAgent windows.

Sign-on

Finger-biometric-based authentication can be used wherever the single sign-on password is used, including the Windows logon, desktop applications, and access to network applications and data. The user can scan its finger as soon as the IBM Security Access Manager for Enterprise Single Sign-On Welcome window opens. The user is provided more detailed instructions if the initial attempt to capture the fingerprint is unsuccessful, as shown in Figure C-50 on page 421.

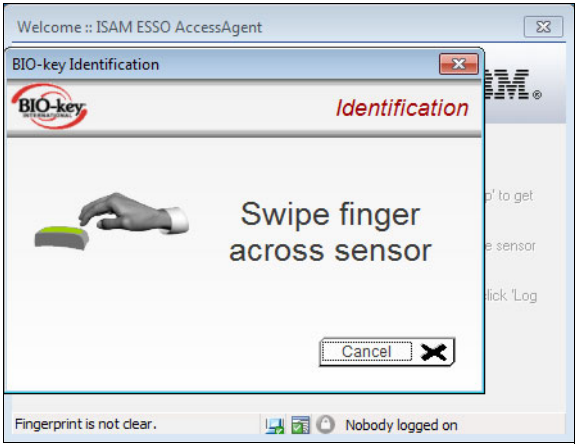

Figure C-50 BIO-key Biometric Service Provider image capture on unsuccessful scan

After the user is successfully logged in to AccessAgent by using an enrolled fingerprint, the user can log on to Windows. IBM Security Access Manager for Enterprise Single Sign-On provides a custom Windows Graphical Identification and Authentication (GINA) (for Windows XP) and Credential Provider (for Windows 7). With the GINA or Credential Provider option enabled in AccessAgent, the user logs on to that service first with a fingerprint. AccessAgent then completes an auto-logon into Windows.

At the Windows desktop, the authenticated user can access desktop applications, and IBM Security Access Manager for Enterprise Single Sign-On provides automated sign-in to network applications and databases, as permitted by the user's Profile.

The specific authentication requirements for access to these resources, however, are defined by the administrator through a set of configurable policies. These policies determine when and where the registration templates cached in the user's AccessAgent Wallet (if implemented) and stored in the IMS Server database are used.

Two separate machine-level policy options can be configured to allow for immediate logon to the Windows desktop. The Fast Logon policy operates in a similar manner to an offline logon process. When this policy is enabled (which it is, by default), IBM Security Access Manager for Enterprise Single Sign-On first uses the locally cached registration template, if available, for authentication. After AccessAgent connects to the desktop, a second machine-level policy option, Background Authentication, triggers a "background" check against the registration template on the IMS Server to detect a revocation or re-enrollment update of the template. These machine-level policies (which means that they apply to all users on that machine) allow the administrator to configure the authentication mode to support any one of three enterprise profiles:

- Users only occasionally re-enroll their fingerprints and create new registration templates, or existing registration templates are only deleted occasionally (by the administrator or Help Desk) when user privileges are revoked.
- Re-enrollment or revocation happens frequently (frequent re-enrollment can happen in settings where users have access to multiple machines and might re-enroll from any of those machines).
- Re-enrollment or revocation never occurs.

For the typical enterprise usage profile where fingerprints are re-enrolled or templates revoked only occasionally (Profile 1), IBM Security Access Manager for Enterprise Single Sign-On provides an authentication mode for access to applications and data, even when the network is slow.

In an enterprise setting where user fingerprints are frequently re-enrolled or existing templates are frequently deleted (Profile 2), it is prudent to immediately validate the fingerprint against the IMS Server and the locally cached fingerprint. IBM Security Access Manager for Enterprise Single Sign-On can be configured to require a match of the reference template against both registration templates for a user before authenticating the user. The local caching of the fingerprint eliminates the need for the user to enter the user name for logon.

If the enrolled user is logging in to the workstation for the first time, however, and the IMS Server is unavailable, the user cannot be authenticated. The user is not allowed access to even the Windows desktop, because the cached fingerprints for that user are not yet available for matching.

Finally, in the uncommon enterprise setting where fingerprints are never re-enrolled or existing enrollments are never revoked (Profile 3), the administrator can allow access to both the desktop and all applications. This access is based solely on a positive match against the locally cached registration template, without a check of the IMS Server - even when the network and IMS Server are available. (This access is allowed by enabling Fast Logon and turning off Background Authentication.) With this authentication mode, revoked reference templates are never removed from the cached Wallet, because there is no IMS Server validation.

Figure C-51 on page 424 illustrates a typical process flow for finger biometric-enabled authentication at sign-on through IBM Security Access Manager for Enterprise Single Sign-On (ISAM E-SSO).

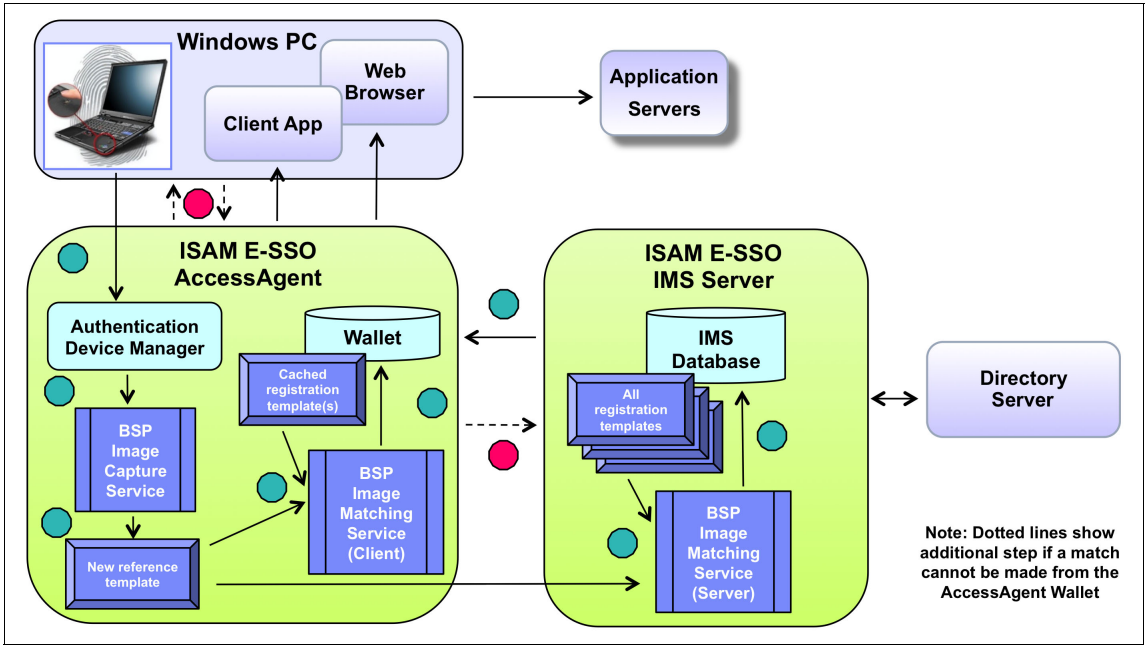

Figure C-51 Typical finger biometric process flow

In this example, the enterprise cached the registration template captured by the BIO-key Biometric Service Provider (BSP) image capture service in the user's AccessAgent Wallet on the workstation and stored it on the IMS Server database. This process flow represents either Profile 1 or Profile 2, in which the BSP image matching service checks against both registration templates.

The authentication process includes the following steps:

1. The user initiates the authentication process by placing a designated finger on the reader attached to or embedded in the workstation.
2. The scanned image is sent to the BSP image capture service.
3. The BSP image capture service creates a reference template.
4. The BSP image matching service in Access Agent compares the reference template against the registration templates cached in the AccessAgent Wallet.
5. A positive match result is reported to the Wallet where it is associated with the user's Profile. With Fast Logon enabled, the user can immediately access the desktop.
6. If there is no match to any cached template (which typically occurs when the user is not at the user's personal workstation or one normally shared), AccessAgent prompts the user for the user name and transmits the user

name with the reference template to the IMS Server for authentication. If the IMS Server is unavailable, the user cannot be authenticated.

7. The reference template is matched with the registration template associated with the user name provided by the Directory Server and stored in the IMS database by the server-level BSP image matching service. With Fast Logon and Background Authentication enabled, this step is deferred until the user is logged on to the desktop.
8. A positive match result is reported to the IMS database where it is associated with the user Profile.
9. Authentication is reported by the IMS Server to AccessAgent on the workstation, which then provides access to the appropriate network applications and databases in accordance with the user Profile.

In the next section, we walk through a detailed configuration for the finger biometric authentication.

Configuring authentication to use fingerprint recognition

This section explains how to configure IBM Security Access Manager for Enterprise Single Sign-On to enable the use of fingerprint readers by using BIO-key Biometric Service Provider (BSP). The BSP SDK must be installed on the machine that runs the IMS Server and on workstations (or other client devices) that run AccessAgent. The versions of SDKs that are run on the server and workstations must be the same. Mixed deployments, although they might work, are not officially supported.

Only administrators can integrate and deploy BIO-key Biometric Service Provider (BSP) with IBM Security Access Manager for Enterprise Single Sign-On.

This section includes the following topics:

- “Prerequisite environment” on page 425
- “Setting up fingerprint authentication” on page 426
- “Updating the user template” on page 437
- “Enrolling the user fingerprint for authentication” on page 440

Prerequisite environment

For platform requirements and configuration instructions, see the IBM Security Access Manager for Enterprise Single Sign-On product documentation.

You need the following resources for this exercise:

- Integrated Management System Server (IMS Server):
  - IBM Security Access Manager for Enterprise Single Sign-On IMS Server prerequisites:
    - WebSphere Application Server
    - IBM HTTP Server
    - A supported database (for example, DB2)
  - Deployment Package for Biometrics (1:1 mapping with an SDK)
  - BIO-key Biometric Service Provider (BSP)
- Client:
  - IBM Security Access Manager for Enterprise Single Sign-On AccessAgent
  - Fingerprint reader
  - Deployment Package for Biometrics (1:1 mapping with an SDK)
  - BIO-key Biometric Service Provider (BSP)

Setting up fingerprint authentication

IBM Security Access Manager for Enterprise Single Sign-On must be configured on the IMS Server (during installation) and AccessAgent (pre-installation). The configuration process includes the following steps:

1. Configuring the IMS Server
2. Configuring AccessAgent
3. Creating and assigning a machine policy template

Configuring the IMS Server

Setting up fingerprint authentication on the server involves installing the Native Library Invoker (NLI) resource adapter on WebSphere Application Server and installing the SDK. The IMS Server installer does not automatically deploy the NLI resource adapter to the WebSphere Application Server. The NLI resource adapter must be installed on every node in the WebSphere cluster. Follow these steps for each node in the cluster:

1. Install BIO-key Biometric Service Provider. Use the manual option provided by the installer GUI to enable support for specific, deployed fingerprint readers only to avoid unnecessary memory consumption on the server.
2. Select **Start** → **All Programs** → **IBM WebSphere** → **Application Server <*version*>** → **Profiles <*profile name*>** → **Administrative console**.
3. Log on to the IBM Integrated Solutions Console, as shown in Figure C-52 on page 427.

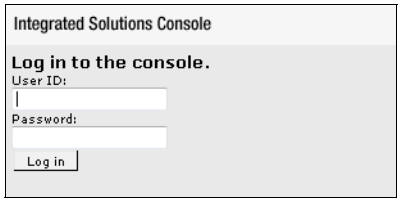

Figure C-52 Integrated Solutions Console logon

4. Install the IBM Security Access Manager for Enterprise Single Sign-On NLI Resource Adapter. This package contains the connector that communicates with the BIO-key Biometric Service Provider to verify fingerprints:

   a. In the Integrated Solutions Console left navigation pane, as shown in Figure C-53, select **Resources** → **Resource Adapters** → **Resource adapters**.

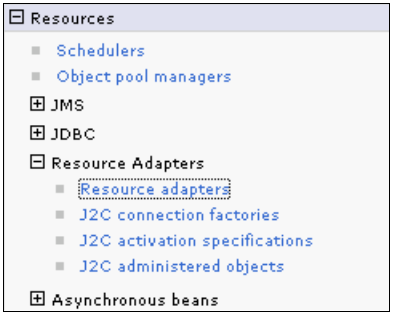

Figure C-53 Selecting resource adapters

   b. On the Resource Adapters summary page, click **Install RAR**. The Install RAR File page is displayed.

   c. Under Scope, select the node on which to install the NLI RAR file. Under Path, select **Local file system** and then provide the full path to the `com.ibm.tamesso.ims-delhi.j2c.adapters.win32.rar` file. This file is in the folder on which the IMS Server is installed (for instance, `C:\Program Files\IBM\ISAM ESSO\IMS Server\`), as shown in Figure C-54 on page 428.

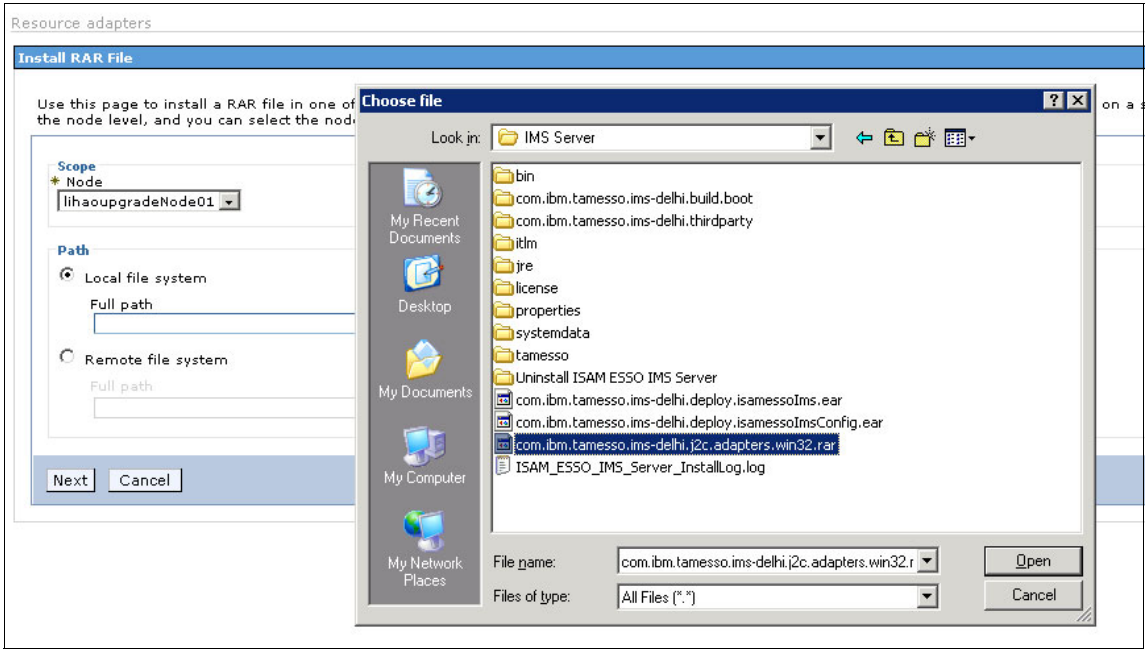

Figure C-54 Finding RAR file name

d. Click **Next**. The General Properties of the new resource adapter are displayed. Keep the default values and click **OK**.

e. In the Messages box at the top of the page, click **Save**. The Resource Adapters summary page is shown again with the newly installed RAR, as shown in Figure C-55.

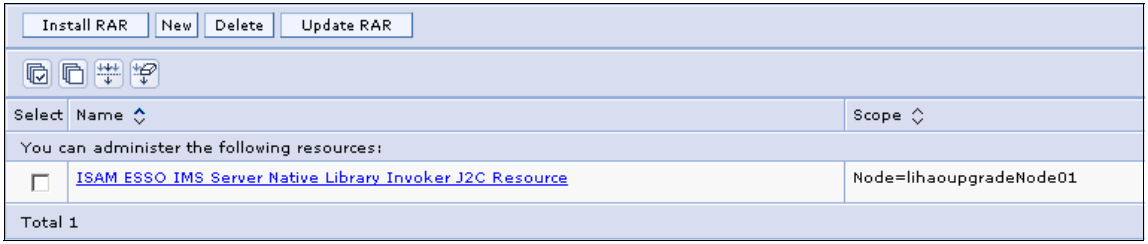

Figure C-55 Updated Resource Adapters summary page

5. Configure the NLI Java 2 Platform, Enterprise Edition (J2EE) Connector architecture (J2C) connection factory:

   a. In the Integrated Solutions Console left navigation pane, as shown in Figure C-56 on page 429, select **Resources** → **Resource Adapters** → **J2C connection factories**.

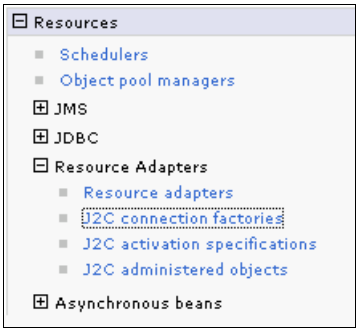

Figure C-56 J2C connection factories

b. In the J2C Connection Factories summary page, click **New**. The General Properties page is displayed, as shown in Figure C-57.

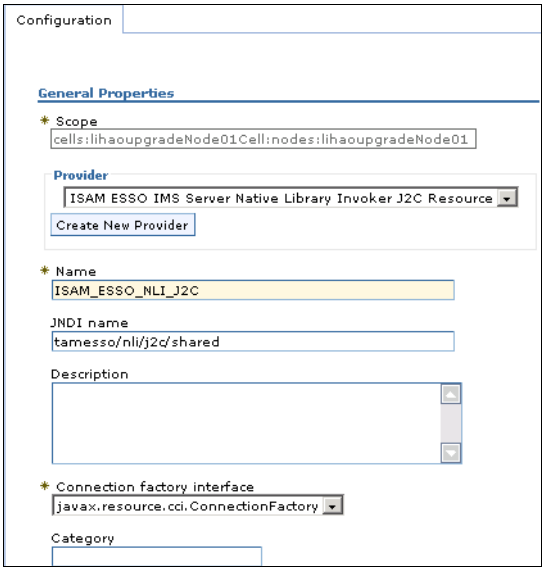

Figure C-57 Configuring the J2C Connection Factory

c. Enter `ISAMESSO_NLI_J2C_ConnFactory` in the Name field.

d. Enter `tamesso/nli/j2c/shared` in the Java Naming and Directory Interface (JNDI) name field.

e. Retain the default values for the other fields. Click **OK**.

f. Click **Save** in the Messages box at the top of the page. The J2C Connection Factories summary page is shown again with the newly created connection factory, as shown in Figure C-58 on page 430.

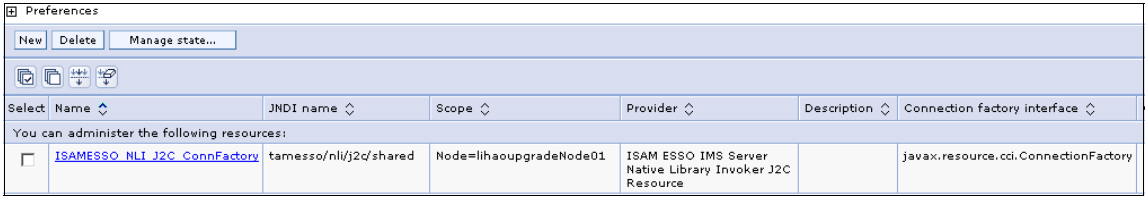

Figure C-58 Updated J2C Connection Factories summary page

6. Run the ISAM ESSO Biometric Deployment Package:

   a. Open the **deploymentPack_biometrics_8.2.0.0.*nnn*** directory from the installation CD (*nnn* represents the version), as shown in Figure C-59.

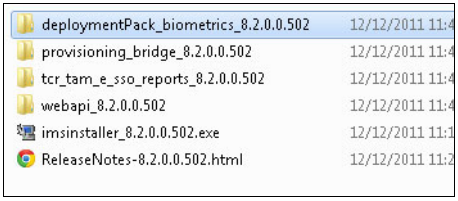

Figure C-59 Opening the installation package

   b. Open the **bio-key** subdirectory, as shown in Figure C-60.

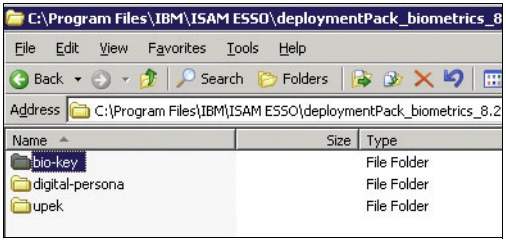

Figure C-60 Locating BIO-key Biometric Service Provider subdirectory

   c. Review the instructions in the `README.txt` file and run `EnBioKeyCOM.bat`.

> **Use the necessary privileges:** The batch file must be run as an administrator. It might be necessary to run the file from a command prompt with sufficient privileges.

   d. After the batch file executes successfully, restart WebSphere Application Server.

Configuring AccessAgent

BIO-key support in AccessAgent requires minor changes to the default installer package, as described in these steps:

1. Install BIO-key Biometric Service Provider. Choose **manual** mode to select specific fingerprint drivers and modules, such as Authentec, to reduce memory consumption on the machine.
2. Open the AccessAgent Installer package. The three folders are Config, Reg, and Customization.
3. Go to the Customization folder.
4. Copy the file `FP3-BioKey.reg` to the Reg folder.
5. Go to the Config folder and open `SetupHlp.ini`.
6. Set ResetBioAPIPermissions to `1`. This property is essential to enable support for the BIO-key EZBioAPI, a module that allows AccessAgent to provide a rich user interface for capture.
7. Now, you can use the installer package to install AccessAgent.

> **Registry settings:** The BIO-key BSP installation creates registry settings in the Local Machine (HKLM) and Current User (HKCU) levels. AccessAgent uses only HKLM settings. HKCU settings are ignored even if these settings are set later by the user.

8. Restart the computer.

If AccessAgent is installed before BIO-key BSP is installed, follow these steps:

1. Go to the Customization folder in the AccessAgent installer package.
2. Run `FP3-BioKey.reg`, which enables support for BIO-key Biometric Service Provider.
3. On a Windows desktop, click **Start** → **Run**.
4. In the Open field, enter `regedit`, then click **OK**.
5. In the Registry Editor, select `HKEY_LOCAL_MACHINE\Software\IBM\ISAM ESSO\SOCIAccess\DSPList\{6EA4B6D4-8CDF4C4E-8B40-CA6A20D0CD6B}\Devices\{5994DB8B-A2C3-4e0a-BC79-F274AE5ECC11}\UISPList\{68F86CB2-630B-4F15-9E2B-5A77B294E9E2}`.
6. Set the registry value Enabled to `1` to enable support for BIO-key EZBioAPI.
7. If the BIO-key version installed is 1.10.*nnn*, go to step 9; otherwise, open the `\Windows\system32` folder.

8. Right-click the folder `BioAPIFFDB`. Set the Security Permissions for this folder to allow Everyone full control. This setting ensures that the BIO-key EZBioAPI is able to run in Guest user accounts.
9. Restart the computer.

Creating and assigning a fingerprint machine policy template

To create and assign a fingerprint machine policy template, follow these steps:

1. Open a browser and enter the IMS Server location. From the IMS Server page, click **AccessAdmin**.
2. Enter the login details for the Administrator for the IMS Server.
3. On the AccessAdmin page, under Machine Policy Templates, click **New Template**. Then, enter the following information, as shown in Figure C-61 on page 433:

   Name — Enter the name of the template. Assign a meaningful name.

   Criteria — Specify the criteria if this template is for specific systems on the domain. Use the default option.

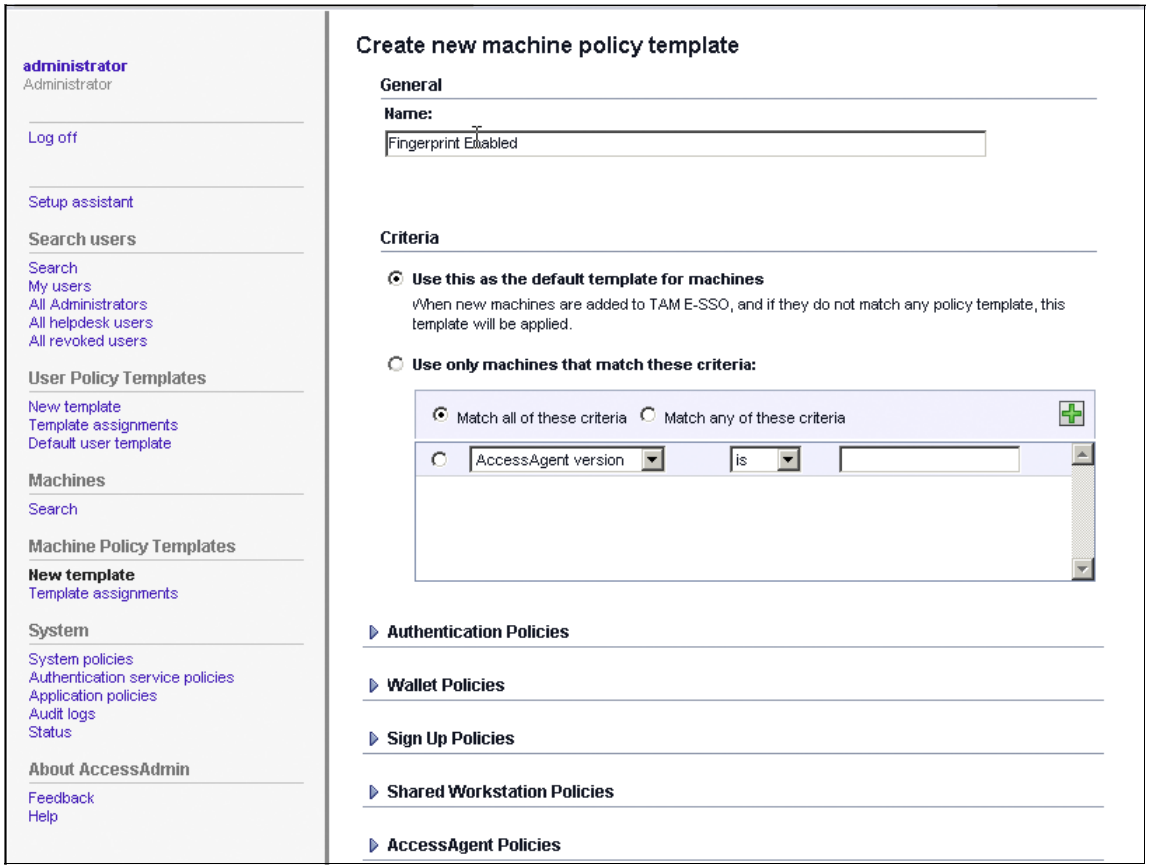

Figure C-61 Entering the template information

4. Under the Authentication Policies section, in the Authentication second factor supported option, enter `Fingerprint` and click **Add**, as shown in Figure C-62 on page 434.

Important: Ensure that you click **Add**. Do not press Enter.

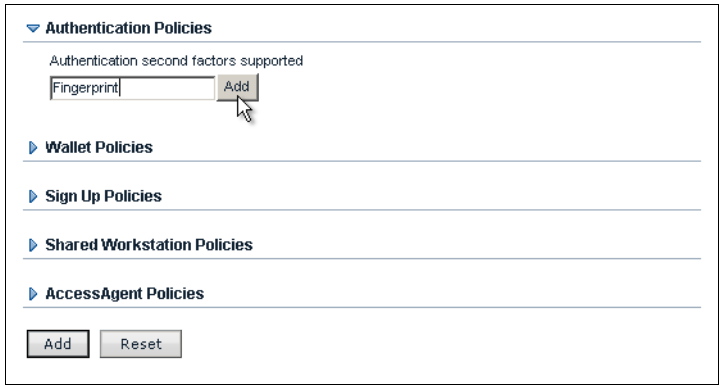

Figure C-62 Add fingerprint authentication policies

5. Scroll down, and add the policy template by clicking **Add**, as shown in Figure C-63.

Figure C-63 Adding the policy template

6. Next, assign this new template to the client system to be used with fingerprint recognition. Go to **Machines** → **Search**, and click **Search** to list the workstations that connect to the IMS Server. Select the workstation.
7. Then, from the Machine policy template assignment section, select the fingerprint policy from the drop-down, and click **Assign** as shown in Figure C-64.

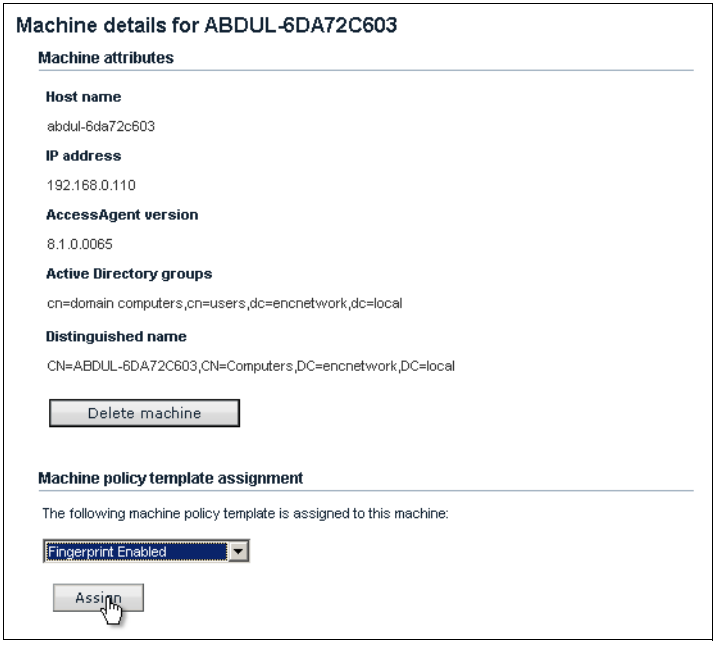

Figure C-64 Assigning the policy template

As illustrated in Table 9-1, specific policies can be set for fingerprint authentication in AccessAdmin.

Table 9-1 Fingerprint authentication machine policies

AccessAdmin policy	Description
pid_fingerprint_tap_same_action	Actions to be performed by AccessAgent when the currently logged on user taps a finger on the reader.
pid_fingerprint_tap_same_action_countdown_secs	Confirmation countdown duration, in seconds, for tapping the same finger on the desktop.

AccessAdmin policy	Description
pid_fingerprint_tap_different_action	Actions to be performed by AccessAgent when a finger is tapped on desktop and it does not belong to the currently logged on user.
pid_fingerprint_tap_different_action_countdown_secs	Confirmation countdown duration, in seconds, for tapping a different finger on the desktop.
pid_fingerprint_registration_max	Maximum number of fingerprints that each user is allowed to register.
pid_fast_logon_enabled	Users with a cached Wallet can log on to AccessAgent without authenticating with the IMS Server. To validate after the logon, you must enable background authentication.
pid_background_auth_enabled_option	Option to specify whether AccessAgent must authenticate with the IMS Server in the background.

For more information about these policies, see the *IBM Security Access Manager for Enterprise Single Sign-On Version 8.2 Policies Definition Guide*, SC23-9694-01.

Updating the user template

Next, specify the user or users who can use fingerprint recognition for authentication:

1. Under the User Policy Template heading, click the **Default user template** link to display the details for the default user policy template.
2. Expand the Authentication Policies heading, and select **Fingerprint**, as shown in Figure C-65. Then, click **Update** at the bottom of the policy template details.

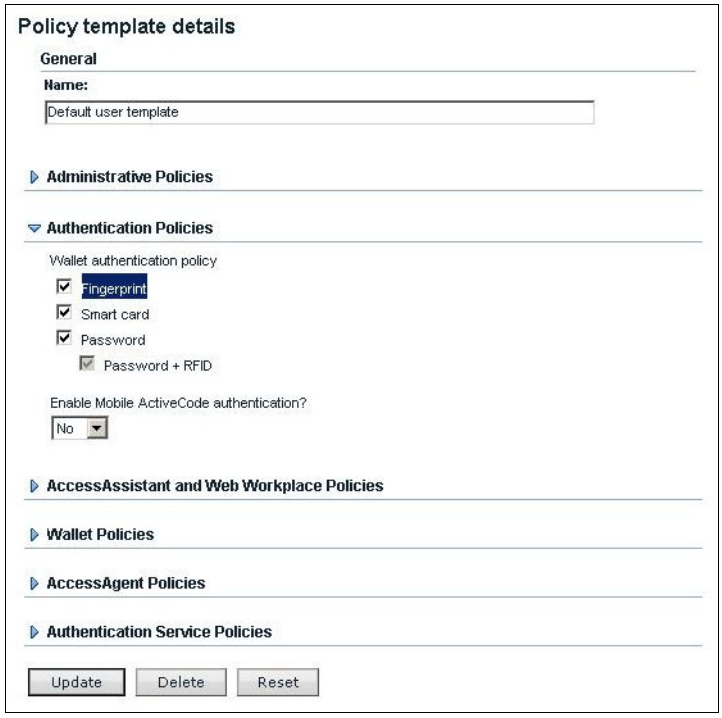

Figure C-65 Updating the user template

3. Click **Search** under the Search for Users heading. Then, click **Search**, as shown in Figure C-66, to list the users who are registered on the IMS Server.

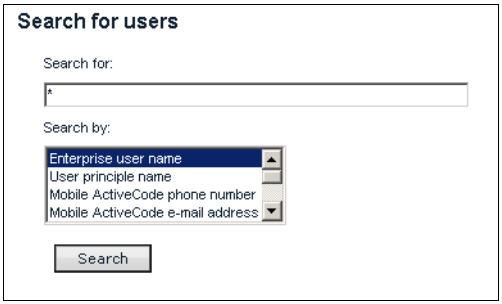

Figure C-66 Identifying users

4. Check the boxes next to the user or users who use fingerprint authentication, as shown in Figure C-67.

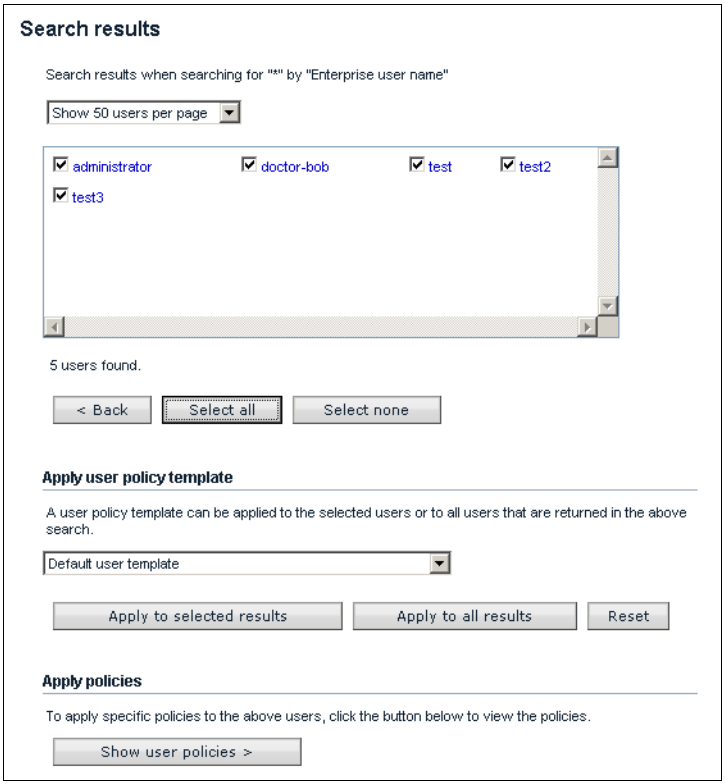

Figure C-67 Selecting users

5. In the Apply user policy template section, expand the drop-down, and select **Default user template**, as shown in Figure C-68. Then, click **Apply to selected results**. When prompted to confirm your action, click **OK** to apply the default user template to the selected users.

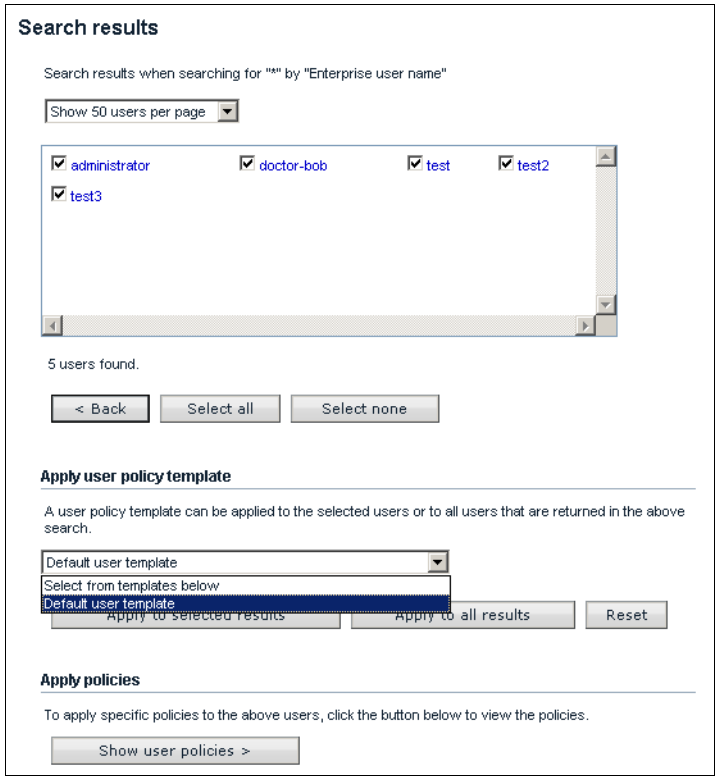

Figure C-68 Selecting the default user template

6. A progress bar displays. When the task completes, restart the client machine. The displayed message on the AccessAgent interface on the client workstation now requests a fingerprint to log on, as shown in Figure C-69.

Figure C-69 Fingerprint requested to log on

Enrolling the user fingerprint for authentication

To enroll the user fingerprint for authentication, follow these steps:

1. Ask the user to scan his or her finger by moving his or her fingertip across the reader. Enter the appropriate user name at the prompt, as shown in Figure C-70. Click **Next**.

Figure C-70 Entering the user name for fingerprint enrollment

2. Select **Register Fingerprint**, as shown in Figure C-71.

Figure C-71 Registering fingerprint

3. Enter the password that is associated with the account, and then click **OK**, as shown in Figure C-72.

Figure C-72 Enter password

4. Identify the scanned finger from the drop-down, as shown in Figure C-73. The maximum number of fingerprint registrations for each user is determined by the administrator.

Figure C-73 Identifying the scanned finger

5. Scan the finger again as directed. As shown in Figure C-74, a message displays when a scan does not result in an acceptable image. BIO-key Biometric Service Provider requires three successful scans. AccessAgent solicits one more scan for verification.

Figure C-74 Scanning user finger

You are logged on to AccessAgent and to the Windows system, as shown in Figure C-76 on page 443. Registration is now complete. You scan only the finger that you registered to log on to your account again.

6. When registration is completed successfully, the fingerprint reference template is stored in the IMS Server and the cached Wallet on the workstation (if configured by the administrator). The user is then presented with the AccessAgent Welcome window with the instructions area muted, as shown in Figure C-75.

Figure C-75 Registration completed

If registering at his or her workstation (as opposed to a dedicated enrollment station), the user can click **Log On** to immediately and automatically access Windows through AccessAgent, as shown in Figure C-76. The AccessAgent icon is visible in the System Tray.

Figure C-76 User logged on to the Windows system

Configuring authentication for Mobile ActiveCode as a one-time password

This section explains how to configure an existing IBM Security Access Manager for Enterprise Single Sign-On environment to use Mobile ActiveCode (MAC) as a *one-time password* (OTP) for non-AccessAgent authentication.

The authentication is done by using AccessAssistant and Web Workplace. AccessAssistant and Web Workplace offer single sign-on without the requirement for an AccessAgent in scenarios where the enterprise applications are web-based. MACs are used to implement second-factor authentication for AccessAssistant and Web Workplace.

Short message service: The steps in this section show how to send the OTP by using a mail server. If required, you can use short message service (SMS) messages instead of emails. You need an SMS gateway and must adapt these instructions as appropriate.

This section includes the following topics:

- "Prerequisite environment" on page 444
- "Creating the messaging connector for email" on page 445
- "Configuring AccessAssistant for MAC second-factor authentication" on page 452
- "Configuring the user account for MAC use" on page 455
- "Logging on with MAC" on page 459

Prerequisite environment

For platform requirements and configuration instructions, see the IBM Security Access Manager for Enterprise Single Sign-On product documentation.

You need the following resources for this exercise:

- Integrated Management System Server (IMS Server):
  - Microsoft Certificate Server
  - Internet Information Services
  - IBM Security Access Manager for Enterprise Single Sign-On IMS Server prerequisites:
    - WebSphere Application Server

    - IBM HTTP Server
    - A supported database (for example, DB2)
  - Smart card middleware

- Mail server

  An email server and client

  Important: The system must be enrolled in the Active Directory domain.

- Client:
  - IBM Security Access Manager for Enterprise Single Sign-On AccessAgent
  - Smart card middleware
  - Initialized smart card and reader or USB token
  - Drivers for reader or token
  - Email client

Creating the messaging connector for email

To create the messaging connector for email, follow these steps:

1. Navigate to the IMS Server, and click the **IMS Configuration Utility** link, as shown in Figure C-77.

Figure C-77 Selecting the connector

2. On the left menu, under Advanced settings, select **Message connectors** to display a drop-down on the right. Select **SMTP Messaging Connector** from the menu, as shown in Figure C-78, and click **Configure**.

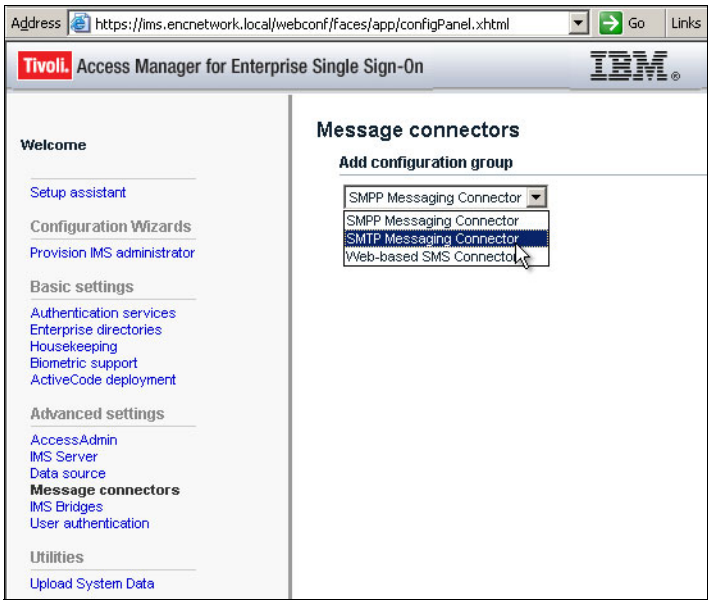

Figure C-78 SMTP Messaging Connector

3. The Simple Mail Transfer Protocol (SMTP) messaging connector enables the delivery of MAC through email. Enter the following parameters, as shown in Figure C-79 on page 448:

Message Connector Name	A name of your choice (for example, `mailServer`)
Address Attribute Name	For example, `emailAddress`
SMTP server URL	For example, `mailServer.`*`domain name`*
SMTP from address	For example, `administrator@mailServer.`*`domain name`*
SMTP form friendly name	Enter a name of your choice (for example, `Admin`)
SMTP port number	`25`
SMTP user name	For example, `administrator@mailServer.`*`domain name`*
SMTP user password	Administrator password

4. After you enter these parameters, click **Add**. The new messaging connector appears.

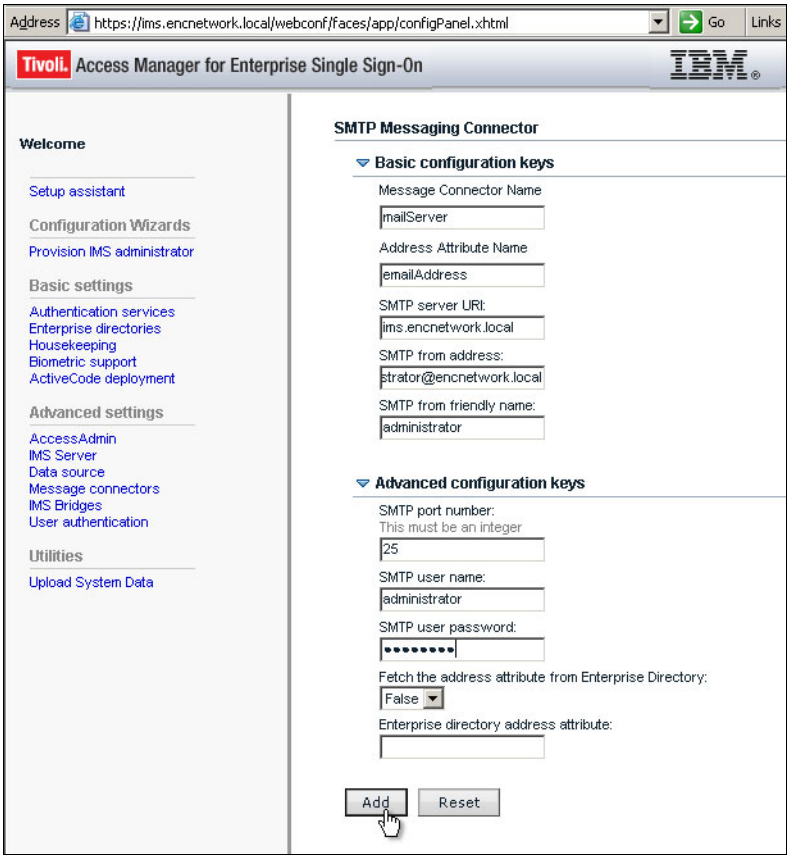

Figure C-79 Entering SMTP messaging connector parameters

5. Click the **ActiveCode deployment** link. Then, to facilitate communication, add the IP addresses of the following systems under Allowed ActiveCode client IPs, as shown in Figure C-80:
   - IMS Server
   - Client Machine
   - Mail Server

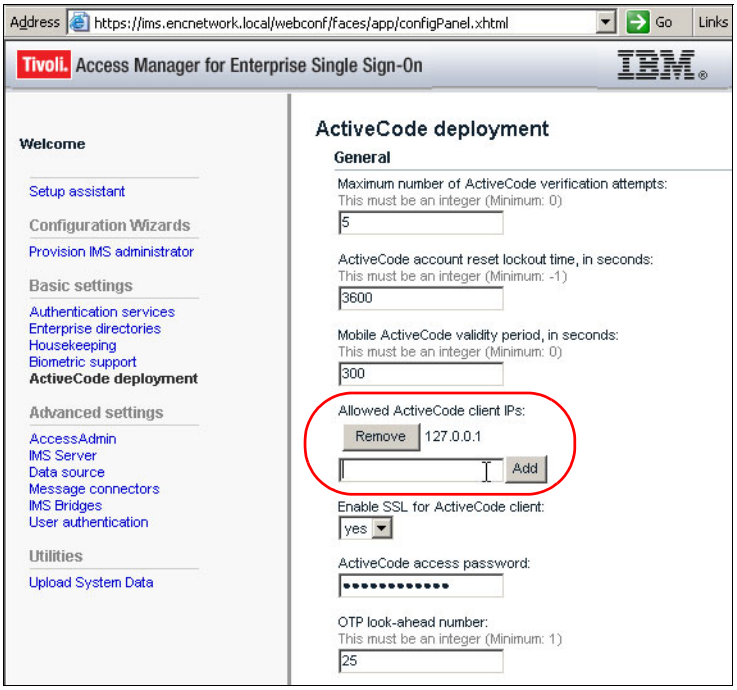

Figure C-80 Adding ActiveCode client IPs

6. Scroll down and enter the name of your SMTP messaging connector in the Default Messaging Connector parameter. Then, scroll to the bottom of the page, and click **Update**, as shown in Figure C-81.

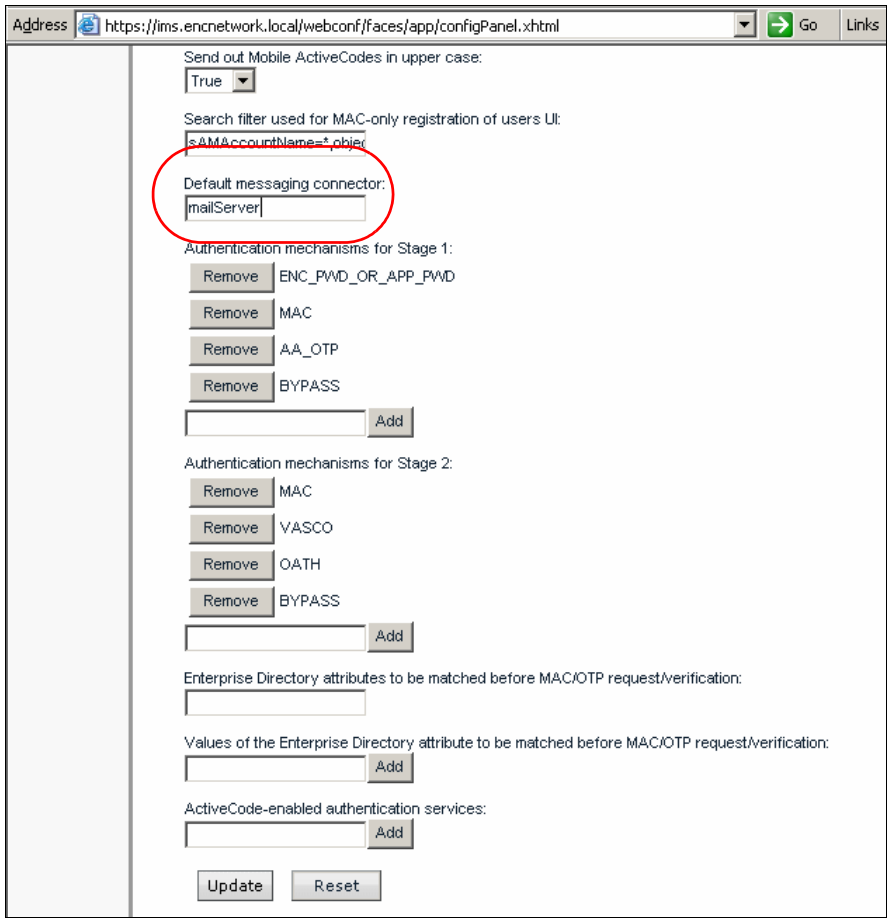

Figure C-81 Setting default messaging connector

7. Restart the IMS Server from the WebSphere Application Server Administrative Console by clicking **Stop** and then by clicking **Start**, as shown in Figure C-82.

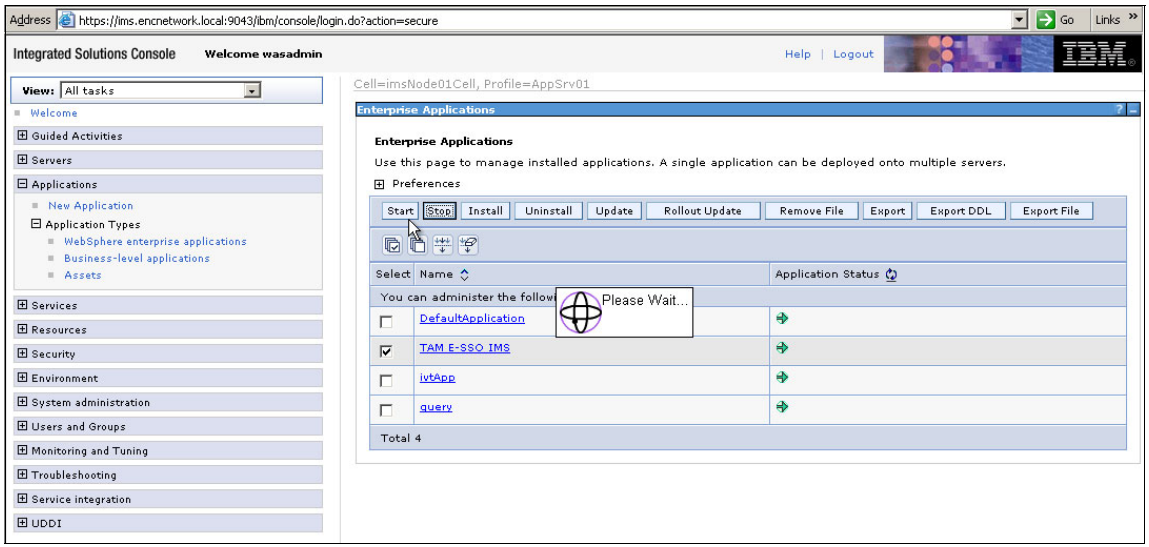

Figure C-82 Restarting WebSphere Application Server

Configuring AccessAssistant for MAC second-factor authentication

To configure the AccessAssistant to use MAC as a second-factor authentication, follow these steps:

1. Navigate to the AccessAdmin page within the IMS Server. On the left menu under the System heading, click the **Authentication service policies** link to list the applications, as shown in Figure C-83.

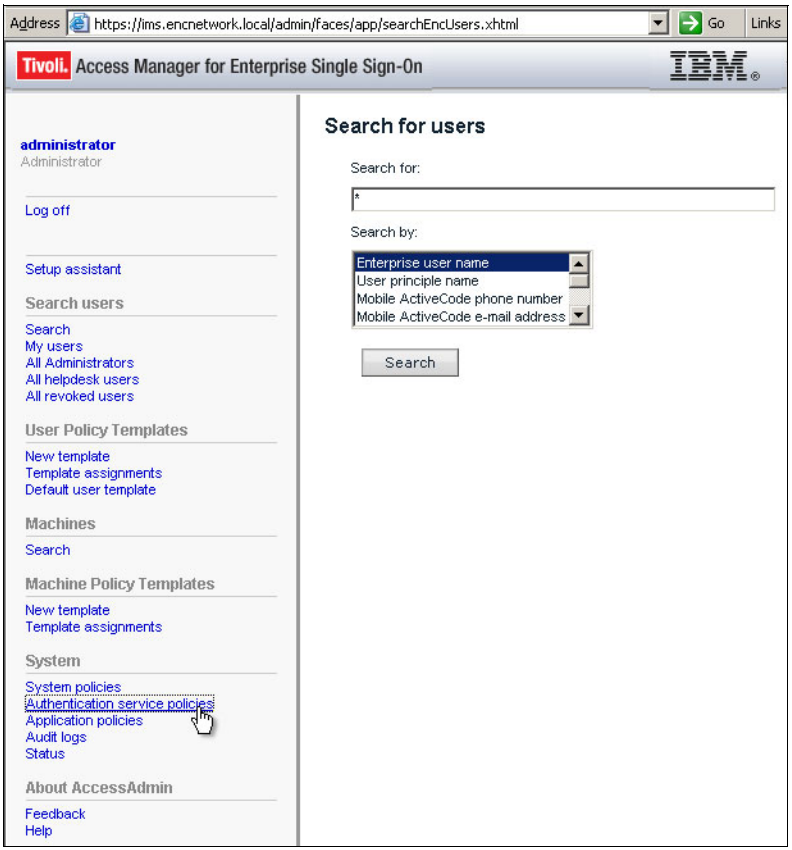

Figure C-83 Selecting Authentication service policies

2. In the Personal authentication services list, select **AccessAssistant** and scroll to the bottom of the page. Click **Move to enterprise authentication services**.
3. Now, under the Enterprise authentication services, click the **AccessAssistant** link, as shown in Figure 4-8.

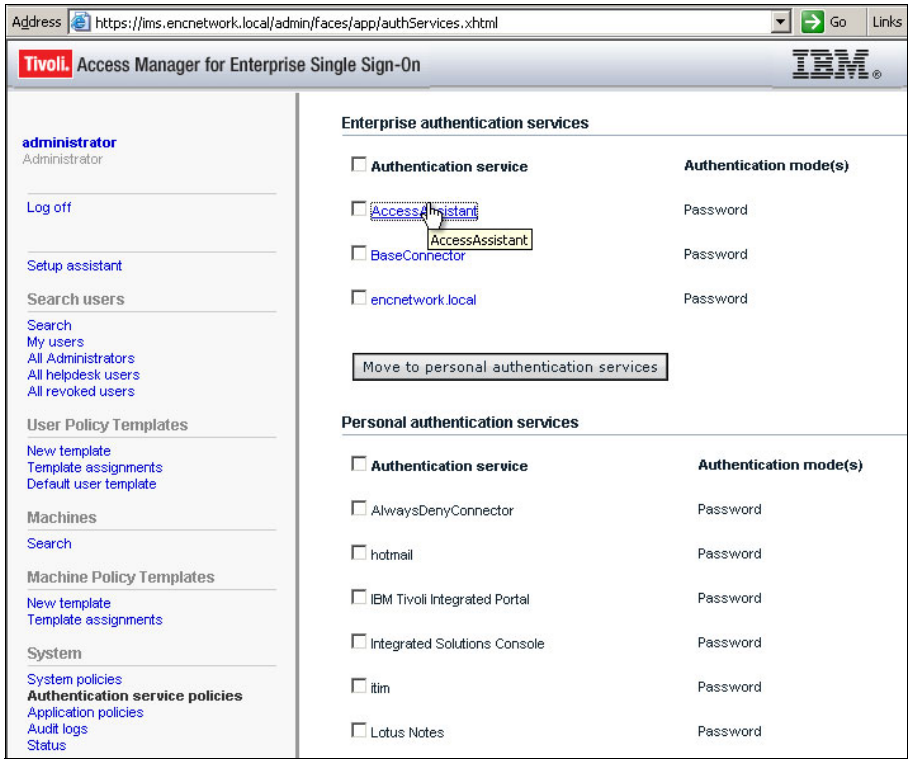

Figure C-84 Selecting AccessAssistant

4. The AccessAssistant Authentication service policies panel appears. Expand the Authentication Policies menu.
5. In the Authentication modes to be supported list, select **Password** and **MAC**, as shown in Figure C-85. Use the Ctrl key for multiple selections. Then, click **Update**.

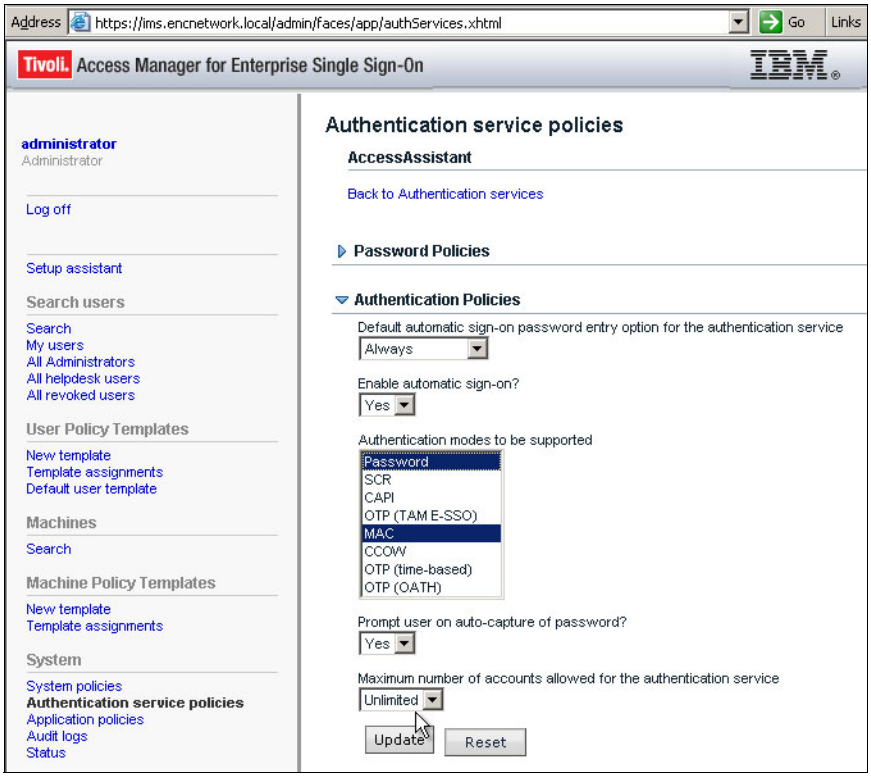

Figure C-85 Selecting MAC and password

6. After you update the information, click the **Back to Authentication Services** link, and move AccessAssistant back to the Personal authentication service list by selecting **AccessAssistant** and clicking **Move to Personal authentication service**.

7. Click the **System policies** link under the System heading in the left pane to display a list of expandable menus. Then, expand the **AccessAssistant and Web Workplace Policies** menu.
8. Select **MAC** from the drop-down list under the Default second authentication factor for AccessAssistant and Web Workplace option, as shown in Figure C-86. Then, scroll down to the end of the page and click **Update**.

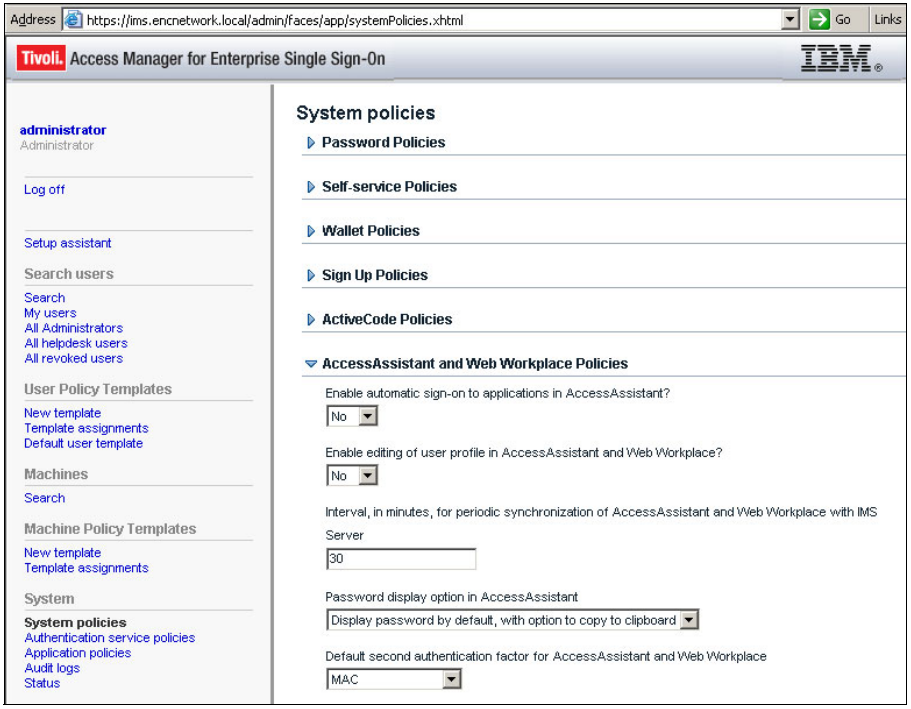

Figure C-86 System policies

Configuring the user account for MAC use

To configure the user account for MAC use, follow these steps:

1. Navigate to AccessAdmin, search for users, and then select the user account for which you want to set up MAC authentication. Enter the following information, as shown in Figure C-87 on page 456:

   Mobile ActiveCode email address Enter the email address of the user to whom the MAC is sent.

Preference Enter the name of the message connector that you created previously (for example, `mailServer`).

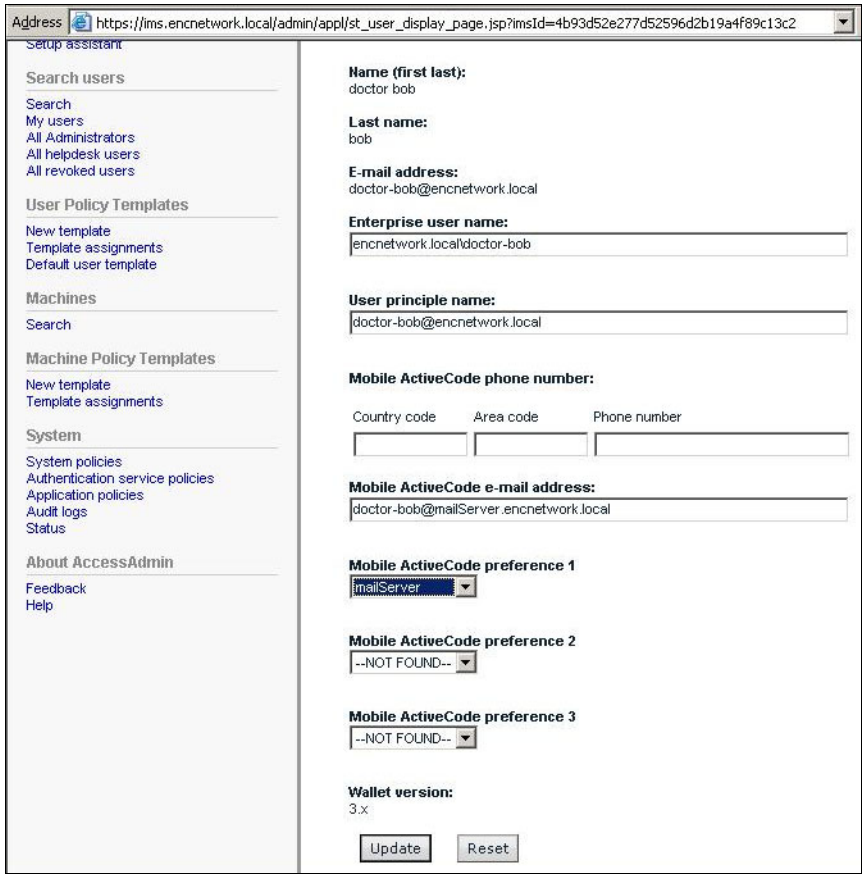

Figure C-87 Entering account information

2. Then, click **Update**.

3. Scroll down to Authentication Policies and enable Mobile ActiveCode Authentication, as shown in Figure C-88, by selecting **Yes** from the drop-down.

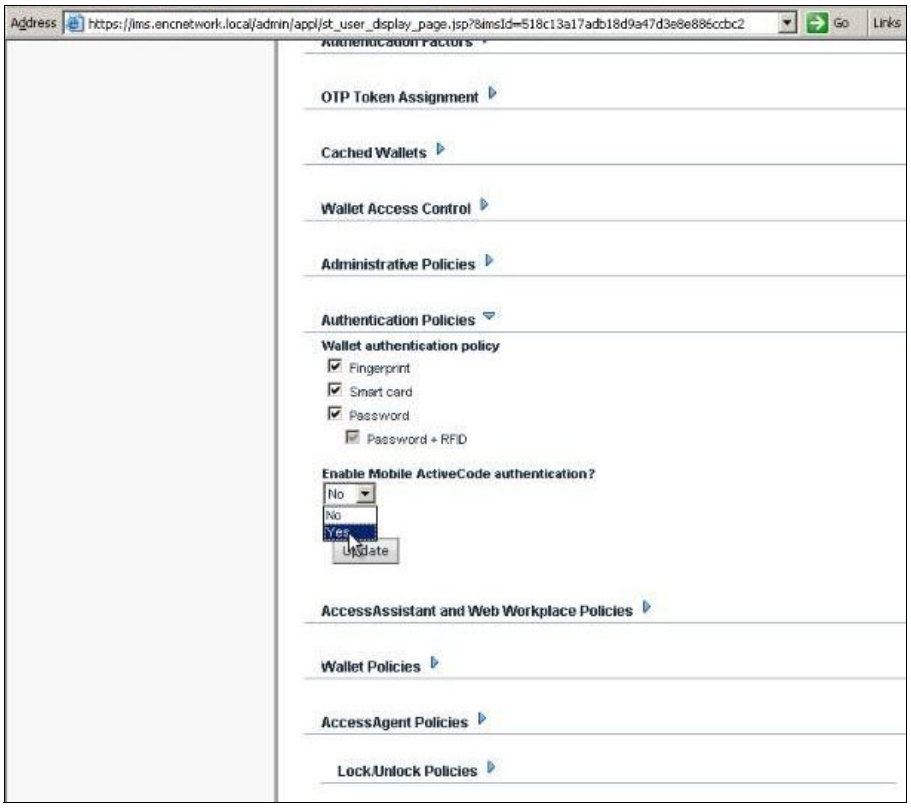

Figure C-88 Enabling Mobile ActiveCode authentication

4. Click **Update**.

5. Now scroll up, and select **Authentication Services**, as shown in Figure C-89.

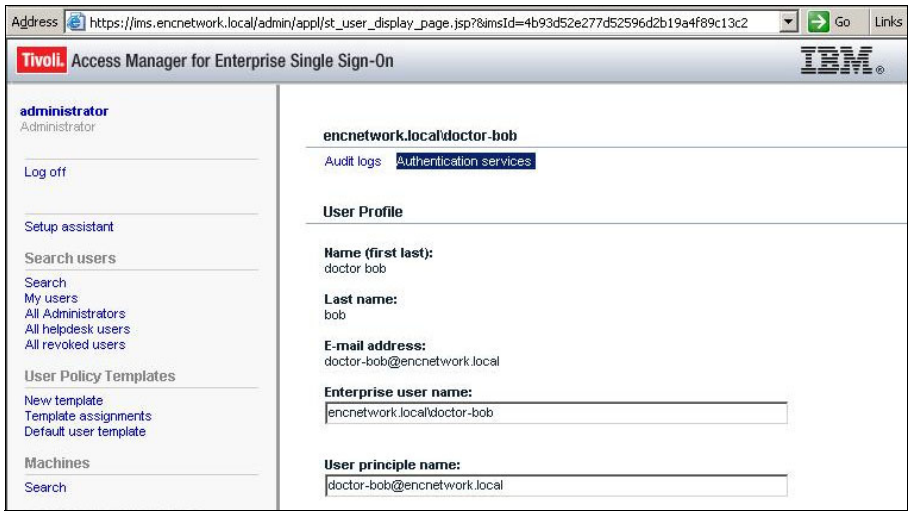

Figure C-89 Authentication services

6. Under ActiveCode-enabled Authentication Services, select the **AccessAssistant** authentication service from the drop-down. Enter the account name to which this service is to be applied (in this case, `doctor-bob`). Then, click **Add account**, as shown in Figure C-90.

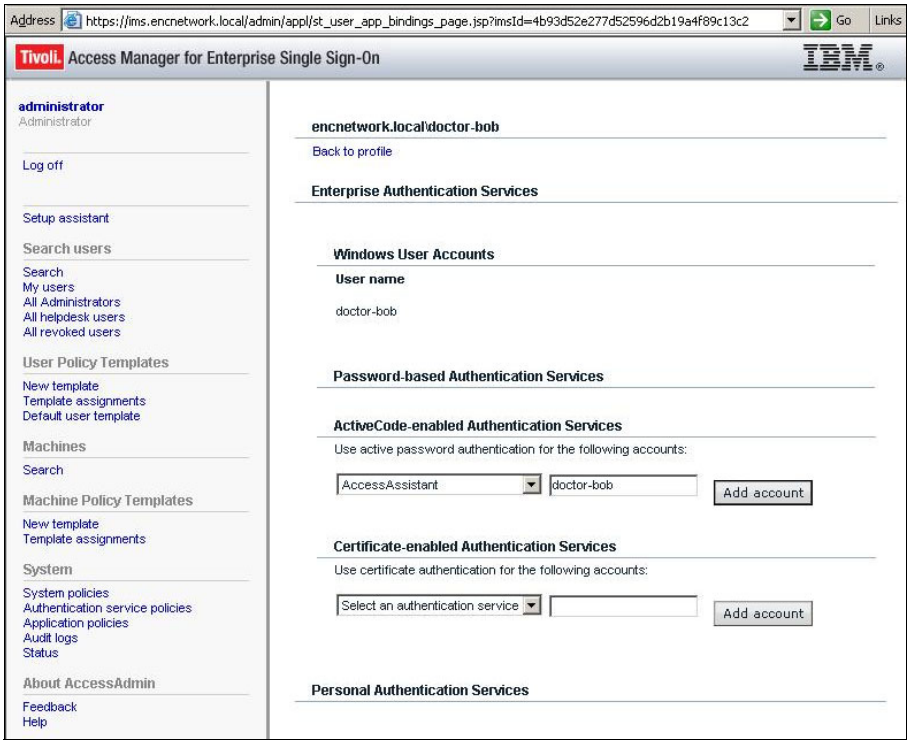

Figure C-90 ActiveCode-enabled authentication services

Logging on with MAC

Follow these steps to log on with MAC:

1. Log on to the Windows system with the user account that is configured to use MAC authentication.
2. Open a browser to access the Web Workplace by using the following URL:

   ims_server_domain_name/aawwp

   If you receive a message about choosing a digital certificate, cancel it.
3. Enter the account details for the user. You are then prompted for a MAC, as shown in Figure C-91 on page 460.

Figure C-91 MAC prompt

4. *Do not close this window.* To find the MAC, go to the email client, and read the relevant email, as depicted in Figure C-92.

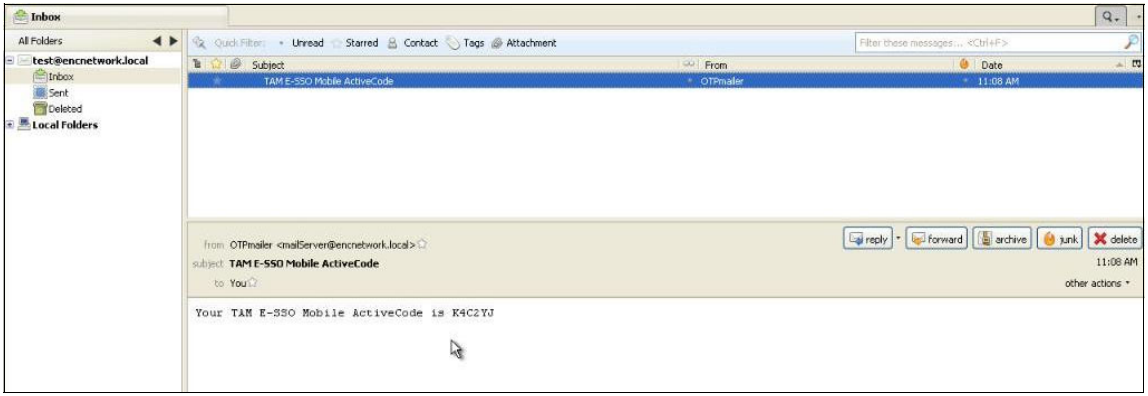

Figure C-92 Obtaining Mobile ActiveCode

5. Enter the MAC into the prompt. You are presented with AccessAssistant and can view your passwords, as shown in Figure C-93.

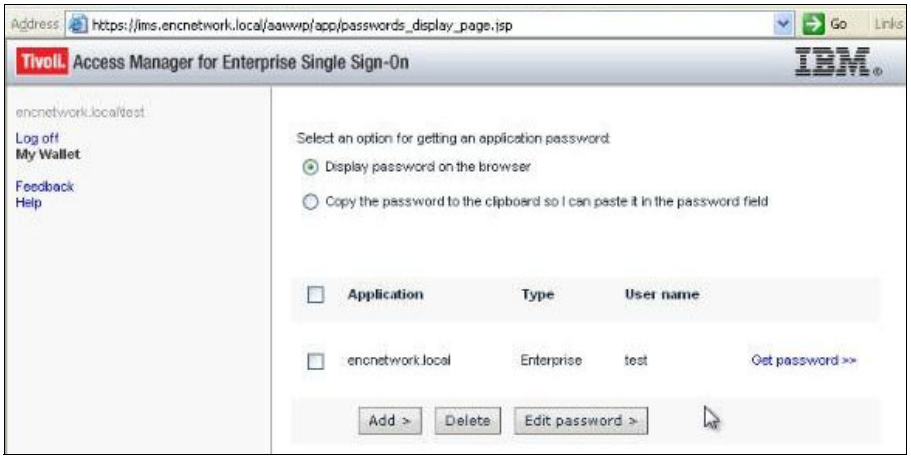

Figure C-93 Obtaining passwords

Conclusion

In this appendix, we demonstrated how to configure and use the following authentication factors with IBM Security Access Manager for Enterprise Single Sign-On, for enhanced security:

- Smart cards
- RFID cards
- Fingerprint readers
- Mobile ActiveCodes (a form of one-time password)

Related publications

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

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 465. Note that some documents referenced here might be available in softcopy only.

- *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014
- *Identity Management Design Guide with IBM Tivoli Identity Manager*, SG24-6996
- *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 Identity Manager 5.0*, SG24-6477

Other publications

These publications are also relevant as further information sources:

- *IBM Security Access Manager for Enterprise Single Sign-On Planning and Deployment Guide Version 8.2,* SC23-9952-03
- *IBM Security Access Manager for Enterprise Single Sign-On Version 8.2 Configuration Guide*, GC23-9692-01
- *IBM Security Access Manager for Enterprise Single Sign-On Administrator Guide Version 8.2*, SC23-9951-03
- *IBM Security Access Manager for Enterprise Single Sign-On User Guide Version 8.2*, SC23-9950-03
- *IBM Security Access Manager for Enterprise Single Sign-On AccessStudio Guide Version 8.2*, SC23-9956-03
- *IBM Security Access Manager for Enterprise Single Sign-On Version 8.2 Policies Definition Guide*, SC23-9694-01

- *IBM Security Access Manager for Enterprise Single Sign-On Version 8.2 Release Notes,* SC22-5402-00

Online resources

These websites are also relevant as further information sources:

- The National Institute of Standards and Technology (NIST) Computer Security Division:

  http://csrc.nist.gov/

  The NIST Computer Security Division is one of eight divisions within the NIST Information Technology Laboratory. The mission of the NIST Computer Security Division is to improve information systems security by performing these functions:

  - Raising awareness of IT risks, vulnerabilities, and protection requirements, particularly for new and emerging technologies
  - Researching, studying, and advising agencies of IT vulnerabilities and devising techniques for the cost-effective security and privacy of sensitive Federal systems
  - Developing standards, metrics, tests, and validation programs:
    - Promote, measure, and validate security in systems and services
    - Educate consumers
    - Establish minimum security requirements for Federal systems
  - Developing guidance to increase secure IT planning, implementation, management, and operation

- IBM IBM Security Access Manager for Enterprise Single Sign-On Support website:

  http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliAccessManagerforEnterpriseSingleSignOn.html?S_CMP=rnav

- The IBM Tivoli Access Manager for Enterprise Single Sign-On Wiki provides preferred practices, education materials, example AccessProfiles, and other documents to enable and support IBM Business Partners, practitioners, and clients with developing AccessProfiles, deploying the product, and learning about the many capabilities of this solution. Subscribe to RSS feeds for the Wiki:

  http://www.ibm.com/developerworks/wikis/display/tivoliaccessmanagerforesso/Home

- IBM Open Process Automation Library (OPAL)

  OPAL is an online ecosystem for sharing Tivoli integrations and solutions developed by IBM and IBM Business Partners (ISVs). It is a comprehensive catalog containing hundreds of predefined integration modules, such as automation packages, adapters, agents, and integration documentation.

  ISVs that list their product integrations in OPAL use OPAL as a go-to-market tool to generate leads and increase their global visibility because clients using OPAL are looking for integrations to enhance the capabilities of their new or existing management applications.

  IBM technical sales teams use OPAL's integrations to support client requirements and proposals. The base OPAL website link is:

  http://www.ibm.com/software/brandcatalog/portal/opal

  The following link takes you directly to the IBM Security Access Manager for Enterprise Single Sign-On page in OPAL:

  http://www.ibm.com/software/brandcatalog/portal/opal/results?catalog.catalogName=Tivoli+OPAL&catalog.searchTerms=&catalog.c=Software_IBM_TivoliAccessManagerForEnterpriseSingleSignOn&catalog.start=0

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

access control
  customization 31
AccessAdmin 10, 33, 37, 201, 206, 221
  password self-service challenge-response questions 201
AccessAgent 9, 12, 33, 46, 108–109
  architecture 17
  cryptobox 62
  installation 149
  interacting with 158
  local user session management 32
  modes 59
  observer agent 26
  observer module 25
  Plug-In 31
  secure storage 62
  server mode 33
  SOAP API 43
  synchronization 110
  Wallet 109
AccessAssistant 41
AccessProfile 13, 18, 26, 109
  central administration 33
  command line application 359
  configuration 168
  creating of 39
  Lotus Notes 169
  shared desktop 32
  storage 44
AccessStudio 18, 33, 39, 108–109
  installation 164
action 29
Active Directory
  lookup user 111
active proximity badge 22
ActiveCode 23
administration
  of deployment 195
Administrative Console
  installation 112
administrative user
  create 111
administrator
  logging 197
advanced profiling 359
audit
  Identity Manager credentials 41
  management 33
  reporting 83
  security 61, 63
audit log
  maintenance 274
auditing
  Tivoli Common Reporting 299
authentication 20
  active proximity badge 22
  ActiveCode 23
  central administration 33
  customization 31
  device manager 20, 24
  factor 12, 18, 20, 64
  fingerprint verification 22
  hybrid smart cards 22
  multi-factor 52, 71
  RFID card 21
  security 61
  service, configuration 168
  smart card 22
authorization code 21
  for password reset 216

B

backup and restore 275
backup password 20
backup site 91, 93
base environment
  configuration 107
behavioral state 27
brute force 5
business
  context 3
  requirements
    shared desktop 260
business requirements 52

C

challenge-response questions 201
Citrix XenApp Presentation Server 12, 32
client-side components 109
communication
  security 62
compliance 55
  deprovisioning credentials 41
component architecture 16
  physical 42
configuration
  AccessProfile 168
  authentication service 168
  base components 111
  base environment 107
  IMS Server 129
  strong authentication 228
corporate security policy 10
cost
  ... of development 54
credential
  distribution 40
    process 8
  request 29
  security 61
  transfer security 30
cryptobox 62
cryptography 6

D

data
  secure processing 61
  synchronization 24
database 36
  administrator 111
deployment
  architecture 109
  considerations 7, 9
  managing 195
  scenario 78
deployment strategies 75
design considerations 108
desktop single sign-on 4
development
  cost 54
dictionary attacks 5
directory 108
directory integration 7
Directory Server
  organization directory 45
disaster recovery 71, 275

E

enterprise authentication services 168

F

fast user switching 32
federated single sign-on 4
fingerprint verification 22
fixlevel 272
functional requirements 56

G

GINA 13, 18
Graphical Identification and Authentication
  See GINA

H

HA (high availability)
  capability 93
help desk
  cost 6, 54
high availability 66, 275
high availability (HA)
  capability 93
HLLAPI 30
hybrid smart cards 22

I

IBM Security Blueprint 13
IBM Tivoli Directory Server
  See Directory Server
identity management
  integration 8
Identity Manager
  credential distribution 40
  integration 47
  password updates 41
  workflow
    extension 47
identity wallet
  See Wallet
IMS
  connector 33
  database 36, 44, 108, 110

provisioning bridge 33
Server 13, 108, 110
configuration 129
installation 112
physical components 43
policy synchronization 24
secure storage 62
SOAP API 33
IMS Configuration Utility 33
IMS database
backup and restore 282
IMS Server
backup and restore 287
scalability 76
incremental rollout 80
information and event management system 63
installation
AccessAgent 149
AccessStudio 164
Administrative Console 112
base components 111
IMS Server 112

J

Java
Observer module 150
Java application
observer agent 31

K

keyboard-sniffing utilities 30
kiosk
single sign-on 8

L

load balancing 69
local user session management 32
log
collection 83
logging 197
logical component architecture 16
logical components
AccessAdmin 37
AccessAgent 17
AccessAgent Observer module 25
AccessStudio 39
authentication 20
data synchronization 24
IMS database 36
provisioning API 40
self-service GUI 25
session management 31
Wallet Manager GUI 24
logon
Mainframe/Host application 30
Web application 30
Windows application 29
lookup user 111, 129
password 112
loss management 33
loss of productivity 54
Lotus Notes
AccessProfile 169

M

Mainframe/Host application
logon 30
Microsoft GINA 18
Microsoft Visual Basic Script 359
Microsoft Windows Server Terminal Services 12, 32
Mobile ActiveCode 23
multi-factor authentication 52, 71

N

Network, Server, and Endpoint domain
scenario 90
security architecture 103

O

observer agent 29
for Java applications 31
for Web applications 30
for Windows applications 29
observer module 18
operational security 61
organization directory 44
overview diagram 12

P

password 20
backup 20
policy 64
reset 21

authorization code 216
functionality 6
scenario 213
security 6
self-service 199
challenge-response questions 201
synchronization 46, 130
updates by Identity Manager 41
performance 7
personal authentication services 168
physical components 42
AccessAgent 46
IMS database 44
IMS Server 43
organization directory 44
physical security 104
policy
management 196
password 64
storage 44, 108
synchronization 110
post-logon 27
pre-logon 27
primary data center 91
primary site 91
private desktop 18, 32, 58
security 65
processing
security 62
product mapping 103
profiling
of applications 359
provisioning
credential distribution 40
provisioning API 40
provisioning bridge 8, 33, 47
proximity badge 22

R

RADIUS API 44
Redbooks Web site 465
Redbooks website
Contact us xiv
regulatory compliance 55
repository 108
requirements
for base installation 109
revocation
of a user 197
RFID
card 21
risk assessment 52
roaming desktop 32, 57, 59

S

scalability 76
scenario
deployment architecture 109
password reset 213
second authentication factors 21
secret 20
secure by design 98
secure storage 61
security 30
AccessAgent 62
architecture of the Network, Server, and End-point domain 103
audit 63
authentication factors 64
communication 62
data processing 62
deprovisioning credentials 41
IMS Server 62
physical 104
policy 10
private desktop 65
requirements 61
storage 104
Wallet 62–63
self-service
challenge-response questions 201
password 199
user interface 25
server-side components 110
session management 31
for local user 32
session management models 56
SetupHlp.ini 150
shared desktop 32, 57
business requirements 260
shared workstation 31, 58
single sign-on paradigm 4
smart card 22
SOAP API 33, 43
solution design 51
state engine

trigger 28
state machine 27
storage security 104
strategies 75
strong authentication 228
strong password 64
system
logging 197
system requirements 109
system security requirements 61

T

target application 110
termination of inactive sessions 8
Tivoli Common Reporting 299
trigger 28

U

user
central administration 33
credentials 13
data storage 108
logging 197
management 196
repository 108
revocation 197

V

virtual appliance 16
high availability 68
virtual desktop
single sign-on 8
Visual Basic Script (VBScript) 359
VMware VDI 59

W

Wallet 8, 13, 18, 109, 275
AccessAssistant self-service interface 41
data synchronization 24
Manager GUI 24
protection 63
revocation 197
secret 20
security 62
Web API 42
Web application
logon 30
observer agent 30
web single sign-on 4
Web Workplace 42
WebSphere Application Server
profile backup 276
Windows
application logon 29
Graphical Identification and Authentication
See GINA
observer agent 29
Terminal Services 12, 32
workflow
automation 31, 39
custom action 31
extension 47

X

XPath 359

Enterprise Single Sign-On Design Guide

Using IBM Security Access Manager for Enterprise Single Sign-On 8.2

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

Enterprise Single Sign-On Design Guide

Using IBM Security Access Manager for Enterprise Single Sign-On 8.2

Holistic design approach for an enterprise single sign-on project

Complete information about architecture and components

Real-world scenario with hands-on details

Everyone feels the pain of too many passwords to remember. Everyone can relate to the security exposure of weak passwords, chosen for convenience. And, everyone can relate to passwords placed in proximity to the workstation for a quick reminder. Unfortunately, that note can allow more than the intended user into the system and network. The average user today often has four or more passwords. And, security policies that focus on password complexity and password-change frequency can cause even more difficulty for users.

This IBM Redbooks publication introduces IBM Security Access Manager for Enterprise Single Sign-On 8.2, which provides single sign-on to many applications, without a lengthy and complex implementation effort. Whether you are deploying strong authentication, implementing an enterprise-wide identity management initiative, or simply focusing on the sign-on challenges of a specific group of users, this solution can deliver the efficiencies and security that come with a well-crafted and comprehensive single sign-on solution.

This book is a valuable resource for security officers, administrators, and architects who want to understand and implement an identity management solution in a medium-scale environment.

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

SG24-7350-02 ISBN 0738437034