

Enabling WebSphere Application Server with Single Sign-on



Configure EIM



Create a SSO enabled Application Server



Prepare and deploy applications



Ursula Althoff
Gary Lakner



International Technical Support Organization

Enabling WebSphere Application Server with Single Sign-on

October 2007

Archived

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

First Edition (October 2007)

This edition applies to Version 5, Release 3, Modification 0 of i5/OS and WebSphere Application Server Version 6.0.

© Copyright International Business Machines Corporation 2007. All rights reserved.

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

Contents

Notices	vii
Trademarks	viii
Preface	ix
The team that wrote this Redpaper	ix
Become a published author	x
Comments welcome	xi
Chapter 1. Introduction	1
1.1 SSO with password elimination	5
1.2 Lightweight Directory Access Protocol	6
1.3 Enterprise Identity Mapping	7
1.4 LTPA mechanism	8
1.5 Identity tokens	9
1.6 Identity Token Resource Adapter	13
1.7 Issues to consider	17
1.7.1 Key timeouts	17
1.7.2 Toolbox connection	18
1.8 Enabling SSO benefits	19
1.9 Introduction of EIM components	19
1.9.1 EIM domain controller	19
1.9.2 EIM domain	21
1.10 Planning work sheets	24
Chapter 2. Enterprise Identity Mapping Configuration	29
2.1 Use the EIM Configuration wizard	31
2.2 Post configuration tasks	43
Chapter 3. Configuring LDAP	45
3.1 Directory Server Web Administration tool	47
3.2 Create the directory database	51
3.3 Templates and realms	53
3.3.1 Create a user template	53
3.3.2 Create a realm	57
3.3.3 Access control lists	58
3.4 Publish SDD data to the directory database	60
3.4.1 Setting up SDD publishing	63

3.5	Create a user for the WebSphere Administrator	65
3.6	Test the directory database	68
3.6.1	Optionally test the connection to the EIM Domain Controller	70
Chapter 4. EIM definitions for SSO with WebSphere		71
4.1	Create an EIM registry definition for WebSphere	72
4.2	Create an EIM identifier	74
4.3	Create associations	75
4.4	Test EIM mappings	80
Chapter 5. Create a new WebSphere Application Server profile provided for SSO		83
5.1	Create a new WebSphere Application Server	84
5.2	Components needed for SSO	95
5.2.1	Start the WebSphere administrator console	95
5.2.2	J2C Authentication Data Entries	97
5.2.3	Identity Token Resource Adapter	98
5.2.4	Connection factories	102
5.2.5	Reinstall resource adapter	108
5.2.6	Trace capabilities of the Identity Token Connection Factory	114
Chapter 6. Enabling your WebSphere Application Server to use single sign-on		117
6.1	Defining the LDAP settings for your WebSphere Application Server	118
6.2	Define the LTPA properties	124
6.2.1	LTPA keys	126
6.2.2	Exporting LTPA keys	128
6.2.3	Importing LTPA keys	129
6.3	Enable Global Security for your WebSphere Application Server	130
6.4	Configure a shared library for the jt400.jar file	133
6.4.1	Create an application class loader	135
6.4.2	Configuring an additional connection factory	138
Chapter 7. Prepare your applications to use single sign-on and EIM		143
7.1	Import an external Connector Resource Archive file into your project	145
7.1.1	Configure the resource adapter to use the EIM domain	147
7.2	Setting up security roles and constraints for your application	151
7.2.1	Define a security role	152
7.2.2	Define a security constraint	153
7.3	Consolidating security roles	158
7.4	Configure authentication settings	160
7.4.1	Configure the authentication settings for the Customer Inquiry application	160

7.4.2 Configure Authentication settings for the Order Entry application (WebFacing)	162
7.5 Define resource reference for both applications	163
7.6 Implement link to the Order Entry application	167
7.7 Export the applications from WDSC	168
Chapter 8. The SSO sample application	171
8.1 How it works	172
8.2 Installing the sample application	172
8.3 Start the sample application	175
8.4 Test the Identity Token sample application	175
Chapter 9. Deploy the Order Entry and Customer Inquiry application ..	179
9.1 Deploy the Order Entry application	180
9.2 Start the new deployed application	187
Appendix A. Single sign-on sample scenario	189
Overview of our scenario	190
Related publications	195
IBM Redbooks	195
Online resources	195
How to get IBM Redbooks	195
Help from IBM	196

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

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

AS/400®
Blue Gene®
CICS®
Domino®
eServer™

i5/OS®
IBM®
iSeries®
OS/400®
Redbooks®

Redbooks (logo) ®
System i™
System i5™
WebSphere®

The following terms are trademarks of other companies:

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

Java, Java Naming and Directory Interface, JavaServer, J2EE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

Preface

There was a time when user names and passwords offered an elegant solution to security concerns. However, this was also the time when companies were just beginning to merge their core processes with technology and had few computer-based applications.

Fast forward to today, when the number of computer applications used daily has surged, and suddenly its elegance has disappeared. Additionally, the popularity of IT solutions (for example, working remotely and Web-based business applications) has increased the demand for secure systems. Consequently, the number of passwords has skyrocketed.

This IBM® Redpaper will walk you through installing and configuring an application on a WebSphere® Application Server that takes advantage of single sign-on. We begin with an overview of Enterprise Identity Mapping (EIM) and how to set it up. Then we set up the application server and enable the applications. The final steps lead you through the process to deploy and use the sample applications provided with the WebSphere Application Server.

The team that wrote this Redpaper

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

Ursula Althoff is working in STG Sales as an IT Specialist for System i™ Technical Sales Germany. She has worked at IBM for 31 years. Her experience on midrange computers started with S/38 and with the System i since its inception. Her areas of experience include i5/OS®, application development, WebSphere Application Server on System i, and WebSphere Development Client for iSeries®. She has developed courses about e-business on System i for IBM learning services, written articles, and has also co-authored several IBM Redbooks® publications about these topics. You can contact Ursula Althoff by sending e-mail to usalthoff@de.ibm.com.

Gary Lakner is a Staff Software Engineer for IBM Rochester on assignment in the ITSO. He is a member of the Blue Gene® Support Team in the IBM Rochester Support Center, where he specializes in both Blue Gene hardware and software, as well as performing customer installations. Prior to joining the Blue Gene team, he supported TCP/IP communications on the IBM eServer™ iSeries server. Gary has been with IBM since 1998.

Special thanks to the following people:

Todd Kelsey
Craig Schmitz
International Technical Support Organization, Rochester Center

Larry Hall
Kevin Lucier
Wayne Ganskop
Chris Smith
IBM Rochester

Davis Marasco
IBM Toronto

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

ibm.com/redbooks

- ▶ Send your comments in an e-mail to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

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

Introduction

There was a time when user names and passwords offered an elegant solution to security concerns. However, this was also the time when companies were just beginning to merge their core processes with technology and had few computer-based applications.

Fast forward to today, when the number of computer applications used daily has surged, and suddenly its elegance has disappeared. Additionally, the popularity of IT solutions (for example, working remotely and Web-based business applications) has increased the demand for secure systems. Consequently, the number of passwords has skyrocketed.

While the user ID and password method itself presents a formidable obstacle to unauthorized access, it is the administration of this solution that is no longer suitable. And employees require a password reset an average of four times per year. Managing multiple user IDs and passwords is risky business - users find it frustrating to remember the combinations and the company finds it costly to force employees to take the time to re-enter this information multiple times each day. The need to streamline IT processes to offer seamless integration between applications, regardless of their platform, is critical to a business' success.

Single sign-on (SSO) is often touted as a solution to reduce or eliminate costs associated with the multiple-password problem. This solution can become very confusing for a couple of reasons. First, SSO is a very ambiguous term and means something different to different people. Second, the typical SSO approaches hide, rather than eliminate, the use of multiple passwords. This

means that the majority of the administrative costs involved in managing user IDs are directly related to managing multiple passwords.

SSO means all of the following things to different people:

- ▶ A user is only prompted once for a user ID and password and never prompted again, no matter what resources on which systems are subsequently accessed by the user directly or through client/server or multi-tier applications.

There are several different approaches that try to accomplish this definition of SSO. None appear to be 100% successful. These solutions often include one or more of the following technologies:

- Synchronizing user IDs and passwords across all user registries.
 - Saving the initial user ID and password provided and sending it to the next tier under the assumption that user ID and password synchronization has been done.
 - Maintaining a cache of user IDs and passwords for each user in each user registry and retrieving the user ID and password from the cache rather than prompting the user for the next tier's user ID and password.
 - Getting the information for the user ID and password cache by screen-scraping the information from the initial prompt for each new system.
 - In multi-tier applications, using a single user ID and password to represent all users authenticated to the middle-tier. This is typically accomplished by either hardcoding the user ID and password in an application or caching the user ID and password in a side file accessible by the application.
- ▶ The user must respond to the same number of prompts, but provides the same information each time.

This view of SSO perceives the problem to be the number of different things a person must manage and remember. In addition, it makes it seem as though the problem is associated with only the user.

The same approaches described in the previous bullet are typically used to address this definition of SSO, except no attempt is made to change applications to remove prompts. Because of this, this definition of SSO can be easier to accomplish.

- ▶ Each user has only one user ID and password stored in one location. This single instance of a user ID and password is used to authenticate users in each security realm.

This definition of SSO perceives the problem to be the number of places that user IDs and passwords are stored. It assumes that if you reduce this number,

you will improve user and administrative productivity and reduce the costs associated with managing multiple user IDs and passwords.

Those who operate under this definition of SSO often assume that a Lightweight Directory Access Protocol (LDAP) server is used to define these user IDs and passwords. Implementing this means that applications and operating system interfaces must be changed to use LDAP for authentication rather than local user IDs defined in an application's or system's user registry. This SSO definition does not address the issues associated with security realms and the enforcement of access control policies within those realms.

This paper describes a different approach to SSO that can be used today in two-tier and multi-tier, heterogeneous applications. This approach is based on the idea of eliminating the need for a system to use passwords associated with local user IDs in order for it to establish which local user ID represents a person or entity making a request. Authentication still occurs, but it does not require that the authentication be done using a local user ID and password. We refer to this as the *password elimination* approach to SSO.

We describe and demonstrate all configuration and implementation steps necessary to exploit the System i5™ password elimination SSO strategy.

First we describe the configuration of EIM and LDAP (an explanation of both of these items are given later on in this document); this functionality is included in i5/OS without additional license costs. We also explain the tasks required to configure a WebSphere Application Server profile (instance) with security enabled to support SSO. Lastly, we describe the steps that have to be done during development and deployment of two different kinds of applications; a WebFacing application and a Web Tools application (developed with WDSC) are shown.

If you want to implement this password elimination SSO strategy for your Web applications, you need to perform the following tasks:

- ▶ Configure Enterprise Identity Mapping (EIM).
See 2.1, "Use the EIM Configuration wizard" on page 31.
- ▶ Install and configure the Lightweight Directory Access Protocol (LDAP).
See Chapter 3, "Configuring LDAP" on page 45.
- ▶ Create EIM registry definitions and EIM Identifiers.
See 4.1, "Create an EIM registry definition for WebSphere" on page 72 and 4.2, "Create an EIM identifier" on page 74.
- ▶ Create Identity Token Resource Adapter and J2C Connection Factories.
See 5.2, "Components needed for SSO" on page 95.

- ▶ Enable and configure WebSphere global security.
See Chapter 6, “Enabling your WebSphere Application Server to use single sign-on” on page 117.
- ▶ Set up SSO for your application.
See Chapter 7, “Prepare your applications to use single sign-on and EIM” on page 143.
- ▶ Deploy your application to the WebSphere Application Server profile.
See Chapter 9, “Deploy the Order Entry and Customer Inquiry application” on page 179.

Note: In this document, we use the new trademarks System i5 and i5/OS. Our descriptions are based on i5/OS V5R3 and WebSphere Application Server Version 6.0 and is also true for iSeries server running i5/OS V5R3 or higher. WebSphere Application Server Version 5.1 also supports SSO in a similar way, which means the concept and components used are the same. The differences are mostly in the area of different panels of the WebSphere Application Server Version 5.1 administration console.

1.1 SSO with password elimination

IBM introduced the Enterprise Identity Mapping (EIM) infrastructure in order to facilitate the implementation of password elimination. IBM also introduced a new authentication mechanism called identity tokens (ID tokens) to make it even easier and cheaper to implement password elimination SSO in multi-tier, heterogeneous applications.

The identity token technology is not really an authentication mechanism. It is an identity assertion mechanism and relies on the concept of third-party trust. This means that trust is established between two applications rather than directly between each application tier and the user who made the original request to the first-tier.

The password elimination approach to SSO is extremely cost-effective, given the business benefits and cost savings. The benefits include:

- ▶ Improved user productivity and satisfaction
Eliminating the use of passwords for authentication between the middle-tier and later tiers implies that password prompts are also eliminated or that it allows the authentication to be done against the same user registry regardless of the system. The cost savings associated with this are relatively small, but they do exist.
- ▶ Reduced administrative work of managing multiple passwords
If passwords on most systems are not used for authentication, they can be removed. This means that you eliminate the majority of the costs associated with managing multiple user IDs and passwords.
- ▶ Reduced help desk and system administration costs

Password elimination (as we implement this with single sign-on (SSO)) is a mechanism where a single user sign-on action permits access to multiple applications that can be running on multiple servers. This allows your Web-based interfaces to access i5/OS back-end applications without having to prompt for additional authentication.

In a strict sense, single sign-on refers to allowing a user to log in to an application with authenticated access to additional applications without encountering additional authentication challenges. In a more real-world sense, it includes mechanisms that can map this primary login into those used for the same person in additional applications.

This can also be made to work in a multi-tier environment where the user uses a Web browser to access middle-tier applications hosted in WebSphere Application Server or WebSphere Portal Server, which then access i5/OS back-end applications. For example, in this environment you can integrate secured WebFacing and Web Tools applications that are configured using single sign-on so that a user only needs to be authenticated once.

This approach's value is twofold:

- ▶ The i5/OS user profile password is not needed to authenticate the user. This means that depending on how other i5/OS user interfaces are accessed by the same users, the i5/OS password can be set to *NONE. This provides the greatest value and potentially a less risky security posture.
- ▶ The System i5 that consumes identity tokens (beginning with V5R3) cuts audit records containing information about the calling application and WebSphere user registry and user ID, along with the local i5/OS user profile that represents the same person, providing an end-to-end audit trail.

So how do we get a WebSphere application and the back-end System i5 application to operate this way? A WebSphere application that needs to connect to a System i5 and gain access to i5/OS data and resources will typically do so using the Java™ APIs provided by the IBM Toolbox for Java or its open-source counterpart, JTOpen. The toolbox communicates with the server through a TCP/IP connection to the i5/OS host servers. Beginning with OS/400® V5R2, both the toolbox and the host servers can accept identity tokens for authentication.

1.2 Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) is a standardized protocol for managing data. Typically, it is used to store and manage the same information stored in a telephone directory, but it is really a generalized, distributed access database and can be used for any kind of information that is relatively static.

Information such as names, phone numbers, and addresses are often stored on a variety of incompatible systems. LDAP provides a simple protocol that enables you to publish this information in a single repository and easily access it over a public or private network. More specifically, LDAP is an open industry standard that defines a standard method for accessing and updating information.

LDAP is also used as a centralized authentication mechanism. In addition to names and phone numbers, passwords can also be stored. When a user or application attempts to access protected information in LDAP, that user or

application must provide a valid user ID and password stored in the LDAP server. LDAP verifies the ID and password.

In the scenario described in this document, LDAP is used for two different purposes. First, it is used to authenticate users to the Web-based application. Second, LDAP is also used as a repository of information representing people that use the Web-based application and the various user IDs that represent that same person; this is known as user identity mapping.

This second use of LDAP is relatively transparent to the application and to the administrator. The application uses Enterprise Identity Mapping (EIM) application programming interfaces (APIs) to query these identity mapping relationships. The EIM APIs do the LDAP interactions, and thus LDAP is really hidden from the application. For simplicity's sake for this scenario, we use a single LDAP server for authentication and as the repository for the EIM information. There is no requirement to do so. We could have just as easily used two different LDAP servers.

To avoid single-point-of-failure issues, LDAP servers provide a replication capability. This enables you to keep multiple LDAP servers synchronized with the same information. If one fails, the other one is still available.

Note: For more information about LDAP, refer to the following IBM Redbooks publications:

- ▶ *Understanding LDAP - Design and Implementation*, SG24-4986, can provide you with some practical guidance.
- ▶ *Implementation and Practical Use of LDAP on the IBM eServer iSeries Server*, SG24-6193.
- ▶ *Using LDAP for Directory Integration*, SG24-6163.

1.3 Enterprise Identity Mapping

Enterprise Identity Mapping (EIM) is a cross platform solution that involves a wide range of technologies including Kerberos, LDAP, and Kerberos Network Authentication Service. Basically, EIM is a framework provided by IBM that allows the mapping of authenticated users to i5/OS (and application) user IDs.

In the scenario in this Redpaper, a user authenticates to an IBM WebSphere Application Server from a browser. EIM is used to enable SSO between a user at a browser, to WebSphere Application Server-based applications, which, in turn, access the i5/OS-based resources.

Without using the EIM mapping, access to System i5 resources from a Web application is typically accomplished by hardcoding a single System i5 user profile and password for use by the WebSphere Application Server applications. However, this means that all users that are authenticated to WebSphere Application Server and authorized to the WebSphere Application Server applications access the i5/OS resources using a single user profile. This causes a loss of accountability on the System i5 where the data is being stored and accessed.

Our scenario does not require hardcoded user IDs or passwords. It accomplishes this by exploiting the EIM infrastructure. The scenario uses EIM for identity mapping purposes. Given one user ID that represents a person, EIM is used to find another user ID that represents the same person. In other words, EIM is required for mapping the ID used for WebSphere Application Server authentication to the profile used to invoke the back-end application on the System i5.

For this, mapping associations between these IDs are defined in the EIM configuration. The user ID used for authentication to the WebSphere Application Server is the source and the System i5 user profile is the target (see 4.3, “Create associations” on page 75).

While EIM provides identity mapping, it does not provide credential mapping. For that reason, a trust mechanism must be in place so that the target application believes the sending application. One commonly used trust mechanism is Kerberos. However, this document focuses on the use of identity tokens (see 1.5, “Identity tokens” on page 9), since they complement EIM quite nicely.

1.4 LTPA mechanism

Lightweight Third Party Authentication (LTPA) is an authentication mechanism intended for distributed, multiple application server and machine environments. LTPA supports forwarded credentials and single sign-on (SSO).

An authentication mechanism in WebSphere collaborates with a user registry. LTPA requires that the configured user registry be a centrally shared repository such as LDAP. It is responsible for creating a credential known as the LTPA token.

When using LTPA, which is a prerequisite for our SSO implementation, a cookie is created containing the LTPA token and inserted into the HTTP response. The LTPA token contains user information and an expiration time and is signed by keys. When the user accesses other Web resources in any other WebSphere Application Server process in the same domain name service (DNS) domain, the

cookie is sent in the request. The LTPA token is then extracted from the cookie and if the receiving servers share the same keys as the originating server, the token can be decrypted to obtain the user information, which is then validated to make sure that it has not expired and that the user information in the token is valid in its registry. On successful validation, the resources in the receiving servers are accessible after the authorization check.

If the request is between different cells of WebSphere Application Servers, you must share the LTPA keys and the user registry between the cells for SSO to work. All of the WebSphere Application Server processes in a cell (deployment manager, nodes, and application servers) share the same set of keys. If key sharing is required between different cells, export them from one cell and import them to the other. Note the name and extension you specify: you must use this file when you configure single sign-on for any additional WebSphere Application Server administrative domains and for Domino®. See 6.2.3, “Importing LTPA keys” on page 129.

The LTPA token is time sensitive. All product servers that participate in a protection domain must have their time, date, and time zone synchronized. If not, LTPA tokens appear prematurely expired and cause authentication or validation failures.

1.5 Identity tokens

Identity tokens are encrypted pieces of data that represent a trusted user identity for a given server or application. They can be passed along a chain of requesters, from one trusted application to the next. Identity tokens are implemented using Enterprise Identity Mapping (EIM). EIM maintains the relationships between Web users and i5/OS user profiles. The application server creates a token for the servers configured to support identity tokens in this EIM Domain.

Identity tokens are not technically an authentication mechanism, but an identity propagation and assertion mechanism. In other words, identity tokens establish trust between the calling and called applications. The calling application asserts the authenticated user's identity (that is, it sends the user ID). EIM is exploited by the called application to determine that the user ID in the called application's user registry represents the same person as the asserted user ID.

Note: EIM and identity tokens do not use passwords.

To better illustrate how this works, let us examine a sample scenario. Application APP1, running on server S1, has authenticated a user against some user registry, UR1. The user registry is typically an LDAP repository or a local OS but is not restricted to these options. APP1 needs to connect to application APP2, which is running on server S2, and is presumably using a different user registry, UR2 to authenticate. Therefore, APP1 needs to know the correct credentials to authenticate to APP2's user registry, UR2. A credential is an internal product representation of a successfully authenticated client user.

The solution is this: APP1 generates and signs (encrypts) an identity token that represents the current user for the current user registry, UR1. APP1 then sends this token to APP2 as a credential. APP2 verifies (decrypts) the identity token to ensure authenticity. APP2 then queries EIM to determine the correct user out of its user registry, UR2. Assuming EIM can map to an appropriate user, APP2 runs as that user because both applications have been configured to trust the EIM server, and therefore trust each other.

We briefly introduced identity token technology in the preceding paragraphs. Here, we describe in a little more detail how ID tokens work. This description only discusses data contained in the token specifically necessary for establishing trust and asserting the user ID. There is other information included in the token that is used to manage the token itself and to make it easier to find data within the token.

ID tokens contain three segments:

- ▶ User information
- ▶ Token manifest
- ▶ Token signature

An ID token is generated by a middle-tier application (WebSphere Application Server). The middle-tier has already determined which local user ID in its user registry represents the person or entity that made a request to it.

Using ID token programming interfaces (or in the case of WebSphere Application Server, an ID token JCA connector, which we describe later), the middle-tier puts the local user ID and a reference to the local user registry in a buffer. This is the user information portion of the ID token. It puts a reference to itself and to the called application in another buffer. This is the token manifest portion of the buffer. These two buffers are combined and then digitally signed by a private key of a dynamically generated public/private key pair. The digital signature is added to the front of the buffer. This ID token is now complete.

Here is where the ID token technology exploits EIM. Using ID token programming interfaces, the calling application at initialization time generates a public/private key pair. The private key is never stored on disk. EIM is used to *publish* the public

key along with a reference to the application that generated it. Only trusted applications are allowed to publish public keys to EIM. By using EIM to publish public keys, public/private key pairs can be used without having to use digital certificates to publish the public key. Keep in mind that EIM does not do any authentication on its own. If the LDAP server trusts the application, then EIM assumes that it is okay to publish the key.

The next-tier or called application (that is, the application being called by the middle-tier application that generated a token) uses ID token APIs to consume an ID token. The verify ID token API accepts an ID token and parses out the token manifest information to find the application ID (that is, the application reference) of the application that claims to have built the token. The API then uses EIM to find the public key associated with the calling application. The public key is used to verify the signature segment of the ID token. If the signature is verified, the called application can trust that the ID token was, in fact, built by the calling application referenced in the token. The verify ID token API then parses the user ID and user registry information from the user information segment. It uses EIM again; however, this time it is used to perform a mapping lookup operation to find a local user ID in the local user registry that represents the same person or entity as represented by the foreign user ID asserted in the ID token. If all is well and verified, the local user ID is returned as the result of the verify ID token API. The called application now knows that this request is being made on behalf of the person or entity associated with the local user ID. The called application can now use the local security manager to enforce local access control policy.

ID tokens have a couple of other interesting characteristics. First, they are only valid for a single use. When consumed, an ID token is no longer valid. Second, they are only valid for a specific and relatively short period of time. The purpose of both of these characteristics is to make it very difficult and unlikely that a replay attack can be mounted against them (for example, that a rogue application could sniff a token from a wire and then use it to authenticate to the same system to perform a different request on its behalf).

Another characteristic is that new public/private key pairs for an application are periodically generated. The ID token API that generates an ID token handles this transparently to the application that calls the API. It also publishes the new key to EIM. This is done to make it harder for a rogue application to generate ID tokens that appear to be generated by a different trusted application. A new key pair is also generated whenever an application is restarted. Dynamic generation of public/private key pairs means that the private key is never stored on disk. This greatly reduces the chances of a rogue application acquiring a trusted application's private key. By choosing a relatively short key pair time out value, relatively short key lengths can also be safely used - the shorter the key, the better the performance of the algorithms that use them.

Yet another characteristic of an ID token is that it can be delegated by an application to which it is sent. This means that the called application can build a new ID token, which essentially contains the ID token it received, plus a new token manifest and new digital signature, over the original and new contents. This means that an ID token contains the information about the entire path a request has taken through a multi-tier application. An application that uses the ID token API to verify an ID token can write audit or log file entries that contain the entire path plus the original user ID that was authenticated by the first-tier in the application.

As with EIM, i5/OS has built-in integrated support for ID tokens. This means that ID tokens can be used with many of the operating system interfaces provided by i5/OS instead of real user profile names and passwords.

Most significantly, however, is that in Version 5 Release 3 and later, when these interfaces receive an ID token, they cut an audit entry of a new type to the system audit journal. The information in the audit entry is the data described in the previous paragraph. In other words, the audit entry contains the information about all of the tiers through which a request flowed on its way to this system along with the original user ID and user registry that was used to authenticate the request at the initial tier of the application. This can be extremely helpful for complying with various regulations or standards.

The ID token APIs are shipped with i5/OS. This means that you can write your own applications running on a System i5 that use ID tokens as the authentication mechanism.

If you have a stand-alone Java application, you can use the Java APIs to build or verify ID tokens. If you want to use ID tokens between a WebSphere Application Server application and i5/OS, it is much easier to use the ID token JCA connector. This hides most of the ID token details from your WebSphere Application Server application. Your application only needs to call the ID token JCA, which builds and returns an ID token and then passes the token as a parameter to the ID token connection method on an AS/400® Toolbox object.

The biggest advantage of ID token technology is that it decouples the authentication mechanism used to authenticate the user at the initial tier of the application from the authentication mechanism used to propagate that authenticated user between various subsequent tiers of the application. You can easily deploy a WebSphere Application Server application in a WebSphere Application Server server that uses LDAP for authentication and then deploy the same application in a different WebSphere Application Server server that uses a different custom user registry (CUR) to authenticate the user with no changes to the WebSphere Application Server application itself. Decoupling the authentication mechanism within the different tiers of the application means that virtually any authentication mechanism can be used to authenticate the user,

regardless of whether that mechanism is supported on any other tiers of the application.

The scenario described in the following chapters uses ID tokens to authenticate between WebSphere Application Server applications and the back-end System i5. Because they are WebSphere Application Server applications, the programmer only needs to know how to deploy the ID token JCA connector and to call the connector from the application.

1.6 Identity Token Resource Adapter

The only piece left, then, is how to generate an identity token in WebSphere Application Server to give to the toolbox to send to the host servers to authenticate to i5/OS.

While it is possible to generate an identity token directly by calling certain Java APIs provided by EIM, another more industry-standard solution now exists: a Java 2 Platform, Enterprise Edition (J2EE™) Resource Adapter, also called JCA connector in combination with a J2C Connection Factory. The IBM resource adapter, the Identity Token Resource Adapter, provides a J2EE application with a fairly simple and configurable way to obtain a connection to the LDAP server hosting EIM and to generate an identity token. The connector code is stored in resource adapter archive (RAR) files.

There are two options to get the resource adapter in place:

- ▶ Install and configure the resource adapter using the WebSphere Administrative Console.
- ▶ Use the JACL script provided alongside the resource adapter code that ships with i5/OS.

Note: The WebSphere Application Server **wsadmin** tool provides the ability to run scripts. You can use the **wsadmin** tool to manage a WebSphere Application Server V6.0 installation, as well as configuration, application deployment, and server runtime operations. The WebSphere Application Server only supports the Jacl and Jython scripting languages. For more information about the **wsadmin** tool and JACL, see the WebSphere online information center.

One of the resource adapter's features is useful when WebSphere security is not enabled (or not configured). The `IdentityTokenFactory` class can generate an identity token for a specified user, instead of assuming there is already an authenticated subject within the current security context. Normally, when WebSphere security is enabled, the resource adapter generates an identity token for the currently authenticated WebSphere user.

However, sometimes it may be necessary to generate an identity token for an application-specified user instead of the current user. The resource adapter can do this as well. This is useful in a WebSphere Portal Server environment when the administrator does not have WebSphere security enabled. Since users typically must log in to Portal Server anyway, the current user for a given portlet request can be used to generate an identity token. This is a trusted solution because the administrator has given the resource adapter permission to interact with the EIM (LDAP) server by providing the LDAP credentials in the resource adapter properties. The administrator also has control over which applications are configured to access the resource adapter.

Note: This option is not quite as secure as enabling WebSphere security in order for WebSphere to automatically generate an internal Java Authentication and Authorization Service (JAAS) subject for the authenticated user, which is then internally consumed by the resource adapter. That is because WebSphere security is not handling user identity verification; it is instead left up to the application.

In a servlet environment, no reliable method exists for obtaining a user identity from a servlet request, unless your application has specific knowledge of a particular user. It is possible to code directly to the WebSphere security APIs and ascertain if WebSphere has allowed someone to log in to the application, and determine who that someone is. But if you are going to do that, you might as well enable WebSphere security and let the resource adapter handle all of it for you. Enabling and configuring WebSphere security is to your advantage, but in case you cannot get it to work, your portlets can still use the resource adapter to generate identity tokens; they just need to pass in the user ID from the portlet request. There might be other applications that want the ability, for whatever reason, to pass in a particular user ID as well.

Using the resource adapter for SSO also provides another benefit. You can use the inherent object-level security built into i5/OS. SSO is achieved without losing accountability, since user identities are mapped instead of hardcoded. Typically, a WebSphere application hardcodes a user ID and password either inside itself or on a `DataSource`, so that all of the connections to the server run under the credentials of one specific identity, no matter who is accessing the application.

While this approach is simple and easy to manage, it does not allow for System i5 that have user profiles and objects already configured with desired authorities.

That is, if your application allows users MARY and FRED to log in, but only perform work to the back-end System i5 server as user JOE, then all of the access rights that MARY and FRED have to specific resources on the System i5 are ignored. Only JOE's access rights are used. Using EIM to map identities provides more flexibility so that MARY accesses all of her i5/OS resources with her own access rights (or the rights of whichever i5/OS profile to which she's mapped in EIM). Also, if you do wish to have a certain application always access the server as one particular user (in this case, JOE), you can create the appropriate associations in EIM so that MARY and FRED both map to JOE. You still do not need to hardcode a user ID and password in your application or DataSource definition.

With the resource adapter in place, a WebSphere application now has the option of generating an identity token and using it to authenticate to the back-end System i5 server.

Your application now needs two things in order to use identity tokens:

- ▶ A resource reference in the deployment descriptor
- ▶ Code to look up the resource adapter at runtime

The resource reference tells WebSphere that your application needs a Java Naming and Directory Interface™ (JNDI) binding for the resource adapter. The resource reference is part of the application's web.xml deployment descriptor; see also 7.5, “Define resource reference for both applications” on page 163.

A sample of the source code needed to perform the JNDI lookup of the resource adapter and generate the identity token is shown in Example 1-1.

Example 1-1 Sample source code for JNDI lookup of a resource adapter

```
// J2EE classes
import javax.naming.Context; // JNDI
import javax.naming.InitialContext; // JNDI
import javax.resource.cci.Connection; // JCA
import javax.resource.cci.ConnectionFactory; // JCA
// Our specific JCA connector classes (idToken.jar)
import com.ibm.jca.idtoken.ConnectionSpecImpl;
import com.ibm.jca.idtoken.IdentityTokenFactory;
// EIM classes (eim.jar)
import com.ibm.eim.token.IdentityToken;
// Toolbox classes (jt400.jar)
import com.ibm.as400.access.AS400;
import com.ibm.as400.access.AS400JDBCdriver;
```

```

...
// Load the resource adapter that you or someone has defined for your
// application.
// Note that most of this code won't work if the JCA connector has not
// been properly configured in WebSphere.
// Also note that the entire premise here is based on an administrator
// having setup user mappings in EIM.
Context ic = new InitialContext();
ConnectionFactory cf =
(ConnectionFactory)ic.lookup("java:comp/env/eis/IdentityToken_Ref");
// Set the parameters into the connection spec.
ConnectionSpecImpl spec = new ConnectionSpecImpl();
// The source application ID is used to differentiate among multiple
// applications using EIM in the enterprise.
// This can (and probably should) be hardcoded in your application.
spec.setSourceApplicationID("Some Name to Uniquely Identify My
Application to EIM");
// The source instance ID is used to differentiate among multiple
// instances of the same application running in the enterprise.
// It needs to be the same for subsequent accesses in the same
// application, but different for a different instance of that
// application.
// A very simple approach we like to use is to generate a random number
// when the application is first loaded (either in init() or via a
// static variable).
spec.generateInstanceID("Some Indicator");
// Get a "connection" to the LDAP server running EIM.
// This is not a transactional connection.
// It is only used for storing the identity token's public encryption
// keys in EIM.
Connection conn = cf.getConnection(spec);
// For demonstration. If this username comes back null, then something
// isn't configured properly in WAS.
String currentWASUser = conn.getMetaData().getUserName();
// Generate an identity token that represents the current WAS user,
// and store the public encryption keys for it in EIM automatically.
// The keys let the consumer of the token verify its authenticity.
IdentityTokenFactory itf = (IdentityTokenFactory)conn;
IdentityToken token = itf.generateIdentityToken();
conn.close();
// Now that we have an identity token, we can use it to connect to the
iSeries.
AS400 sys = new AS400("serverName");
byte[] tokenBytes = token.toBytes();
sys.setIdentityToken(tokenBytes);

```

```
// And we can perform tasks, such as running a CL command.
CommandCall cc = new CommandCall(sys);
cc.run("CRTLIB FRED");
// Here is how you would use it to get a JDBC connection.
// Since the java.sql spec doesn't define a nice way for us to pass an
// identity token, we have to hardcode the Toolbox driver.
AS400JDBCdriver driver = new AS400JDBCdriver();
// Specify false in order to use the specified AS400 object, not a
clone.
java.sql.Connection jdbcConn = driver.connect(sys, false);
// Now do SQL work with jdbcConn like you normally would.
Statement st = jdbcConn.createStatement();
ResultSet rs = st.executeQuery("SELECT * FROM QIWS.QCUSTCDT");
```

The WebFacing and Web Tools utilities inside WebSphere Development Client for iSeries (WDSC) makes using this connector even easier. Each of the utilities provides you with configuration options that SSO enable these applications by using ID tokens. The appropriate code is generated and added by the utility to generate and pass the ID token to the server for authentication.

1.7 Issues to consider

There are a few issues you may come across when using identity tokens in your application, including key timeouts and how the toolbox connects to i5/OS.

1.7.1 Key timeouts

The resource adapter has a key timeout period, which is the number of seconds that a published identity token key remains valid. After the timeout period the published key is no longer used to sign newly generated identity tokens. Once the timeout is reached, a new key is published that will be used to sign newly generated identity tokens. After a key expires it is used as a backup key until the new current key expires. In essence what this means is that a key timeout of 20 minutes is valid for twice the key timeout, or in this case, 40 minutes.

Note: If no new identity tokens are generated, the published keys remain valid.

1.7.2 Toolbox connection

Also related to the key timeout is a scenario dealing with how the toolbox connects to i5/OS. The toolbox's *AS/400 object* is the Java object that represents the socket connections to the i5/OS host server jobs. There are multiple host servers, including database, file, print, sign-on, central, remote command, and so on. The AS/400 object encapsulates socket connections to each of these host servers, and the toolbox APIs are structured such that connections to the host servers are established implicitly or explicitly. Every time a new socket connection to a host server is initiated, a sign on procedure occurs (for obvious security reasons). In a long-running application, then, a given AS/400 object may have connected to the remote command server an hour ago and may need to connect to the file server an hour from now. This is especially true if you use a caching mechanism or connection pool for your AS/400 objects. For example, Web applications commonly cache the AS/400 object in the HTTP session.

If you give the AS/400 object an identity token, it uses that token for authentication with the host servers. If the timeout value for the key that generated the token is exceeded, such that the key is no longer published in EIM (as either the current or backup key), and the AS/400 object then needs to connect to another host server, the connection attempt will fail because the sign on procedure will be unable to decrypt the identity token. This means that pooled or cached AS/400 objects could become invalid during an application's lifetime. This scenario can be avoided in several ways:

1. Set the key timeout on the Identity Token Resource Adapter to be something more than 20 minutes. Some Kerberos environments, for example, use a time out of eight hours.
2. Create a new AS/400 object every time i5/OS work needs to occur. This could be slow, since connection pooling is not used.
3. Pre-connect all of the host servers that your application may ever need to use. You can do this by calling `AS400.connectService(SERVICE)`, where `SERVICE` is one of the constants on the AS/400 class that represents the different host servers (for example, `AS400.DATABASE`, `AS400.FILE`, `AS400.COMMAND`, and so on). As long as those services are connected and the underlying TCP/IP socket is not dropped, you would not need to re-authenticate that particular AS/400 object to those host server jobs.

Note: This will cause extra, possibly unneeded, host server jobs in i5/OS.

The application developer would have to decide which is better: A longer connection initiation time with fewer server jobs (as in No. 2) or a shorter connection initiation time with more server jobs (as in No. 3). If your application is

using an AS/400 object connection pool or caching mechanism, then technically you have already made this decision.

1.8 Enabling SSO benefits

The Identity Token Resource Adapter, in conjunction with the IBM Toolbox for Java or JTOpen, is a great way to enable your WebSphere and Portal Server applications for SSO to the System i5. Some up-front resource adapter configuration is required, in addition to setting up EIM on an LDAP server and entering all of your enterprise's necessary identities and associations. Once configured, the resource adapter is managed through the WebSphere administrative console, so any application can be assigned its own identity token factory and settings. Mapping authenticated WebSphere identities to back-end System i5 user profiles not only achieves SSO, but it is also beneficial to applications or environments that rely on i5/OS object-level security. EIM and identity tokens are being adopted by various software projects, such as iSeries Access for Web and WebSphere Development Studio Client (WDSC), including the WebFacing Tool.

1.9 Introduction of EIM components

Here we discuss the basic concepts of Enterprise Identity Mapping (EIM). For more detailed information, refer to the System i5 Information Center under **Networking → Network Security → Enterprise Identity Mapping**.

1.9.1 EIM domain controller

The EIM domain controller is an LDAP server that is configured to manage at least one EIM domain. An EIM domain is an LDAP directory that consists of all the EIM identifiers, EIM associations, and user registries that are defined in that domain. Systems (EIM clients) participate in the EIM domain by using the domain data for EIM lookup operations. A minimum of one EIM domain controller must exist in the enterprise.

Currently, you can configure some IBM platforms to act as an EIM domain controller. Any system that supports the EIM APIs can participate as a client in the domain. These client systems use EIM APIs to contact an EIM domain controller to perform EIM lookup operations.

The location of the EIM client determines whether the EIM domain controller is a local or remote system. The domain controller is local if the EIM client is running on the same system as the domain controller. The domain controller is remote if the EIM client is running on a separate system from the domain controller.

Note: If you plan to configure a directory server on a remote system, the directory server must provide EIM support. EIM requires that the domain controller be hosted by a directory server that supports Lightweight Directory Access Protocol Version 3. Additionally, the directory server product must be configured to accept the EIM schema. The IBM Directory Server for iSeries provide this support.

Figure 1-1 shows an EIM implementation.

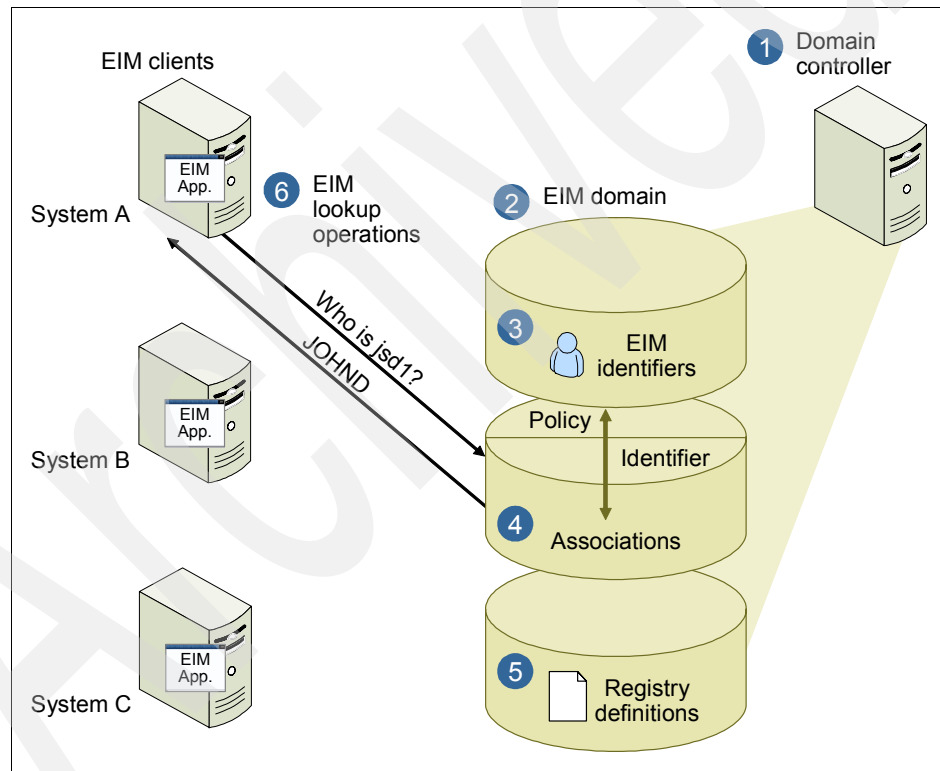


Figure 1-1 An EIM implementation example

1.9.2 EIM domain

An EIM domain is a directory within a Lightweight Directory Access Protocol server that contains EIM data for an enterprise. An EIM domain is the collection of all the EIM identifiers, EIM associations, and user registries that are defined in that domain, as well as access control for the data. Systems (EIM clients) participate in the domain by using the domain data for EIM lookup operations. An EIM domain is different from a user registry. A user registry defines a set of user identities known to and trusted by a particular instance of an operating system or application. A user registry also contains the information needed to authenticate the user of the identity. Additionally, a user registry often contains other attributes such as user preferences, system privileges, or personal information for that identity. In contrast, an EIM domain refers to user identities that are defined in user registries. An EIM domain contains information about the relationship between identities in various user registries (user name, registry type, and registry instance) and the actual people or entities that these identities represent.

Figure 1-2 shows the data that is stored within an EIM domain. This data includes EIM identifiers, EIM registry definitions, and EIM associations. EIM data defines the relationship between user identities and the people or entities that these identities represent in an enterprise.

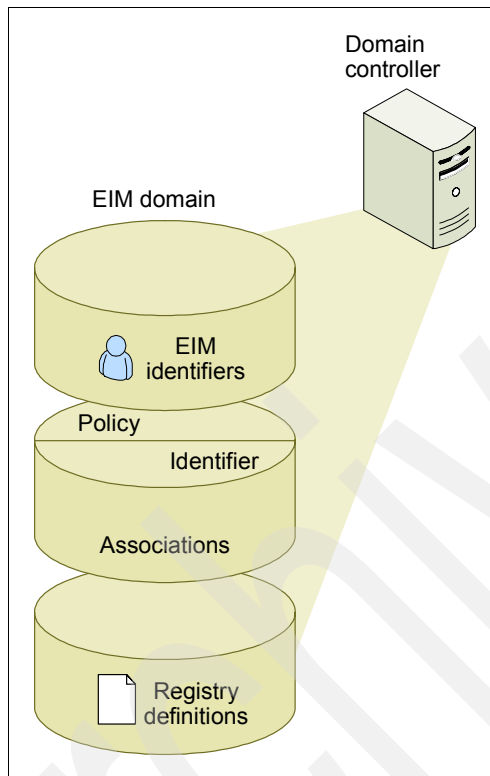


Figure 1-2 EIM domain and the data that is stored within the domain

EIM data includes:

1. EIM registry definitions

Each EIM registry definition that you create represents an actual user registry (and the user identity information it contains) that exists on a system within the enterprise.

Once you define a specific user registry in EIM, that user registry can participate in the EIM domain. You can create two types of registry definitions, where one type refers to system user registries and the other type refers to application user registries.

2. EIM identifiers

Each EIM identifier that you create uniquely represents a person or entity (such as a print server or a file server) within an enterprise. You can create an EIM identifier when you want to have one-to-one mappings between the user identities that belong to a person or entity to whom the EIM identifier corresponds.

3. EIM associations

The EIM associations that you create represent relationships between user identities. You must define associations so that EIM clients can use EIM APIs to perform successful EIM lookup operations. These EIM lookup operations search an EIM domain for defined associations. There are two different types of associations that you can create:

- Identifier associations

Identifier associations allow you to define a one-to-one relationship between user identities through an EIM identifier defined for an individual. Each EIM identifier association that you create represents a single, specific relationship between an EIM identifier and an associated user identity within an enterprise. Identifier associations provide the information that ties an EIM identifier to a specific user identity in a specific user registry and allow you to create one-to-one identity mapping for a user. Identity associations are especially useful when individuals have user identities with special authorities and other privileges that you want to specifically control by creating one-to-one mappings between their user identities.

- Policy associations

Policy associations allow you to define a relationship between a group of user identities in one or more user registries and an individual user identity in another user registry. Each EIM policy association that you create results in a many-to-one mapping between the source group of user identities in one user registry and a single target user identity. Typically, you create policy associations to map a group of users who all require the same level of authorization to a single user identity with that level of authorization.

1.10 Planning work sheets

Here we discuss the planning work sheets.

User and password on System i5

Table 1-1 User and password on System i5

User profiles	User	Password
What are the OS/400 user profile names for these users? <ul style="list-style-type: none">▶ John Day.▶ Web Facing Advanced.▶ Web Facing Original.▶ Ursula Althoff.▶ Dieter Werkmann.	JOHND WFADV WFORG ALTHOFF WEKMANN	SECRET WF400LAB WF400LAB WORK4FUN WORK4FUN
What are the user names / password in the IBM Directory Server (LDAP)? <ul style="list-style-type: none">▶ All System i5 users that have a WRKDIRE, because we activated the function "publish the System i5 data to the directory database", see "Publish SDD date to the directory database" on page 60.	as on System i5	no password set

Directory Server and EIM parameter

Table 1-2 Planning work sheet for configuring a Directory Server and EIM in i5/OS

Component	Value set and used	Initially Set by	Referenced by
EIM Domain controller name	AS270DD.DUEDORF. DE.IBM.COM Port 389	Administrator using iSeries Navigator when configuring EIM, see step 8 on page 34.	
EIM Domain Name and its parent domain	EIM_FFTS Parent DN= dc=AS270DD,dc=DUE DORF,cd=DE,cd=IBM, cd=COM	adminiStrator using iSeries Navigator when configuring EIM, see steps 10 on page 36 and 11 on page 37.	EIMDomainName and ParentDomain

Component	Value set and used	Initially Set by	Referenced by
Local user registry name (LDAP)	AS270DD.DUEDORF.DE.IBM.COM	Administrator using iSeries Navigator when configuring EIM see step 12 on page 38.	
Directory suffix	Parent DN=dc=AS270DD,dc=DUE DORF,cd=DE,cd=IBM,cd=COM	See 3.2, "Create the directory database" on page 51.	
LDAP administrator distinguished name (DN) and password	cn=administrator Password = work2win	Administrator using iSeries Navigator when configuring EIM, see step 9 on page 35.	J2C Authentication Data entry (WebSphere Application Server configuration)
LDAP Console Administrator when using Directory Server Web Administration tool	administrator	Set up the Directory Server Web Administration Tool, see "Configure the Directory Server Web Administration Tool" on page 48.	
Template for employees	cn=employees Parent DN=dc=AS270DD,dc=DUE DORF,cd=DE,cd=IBM,cd=COM	Create a user template; see 3.3.1, "Create a user template" on page 53.	

EIM Identifier parameter

Table 1-3 shows the EIM Identifiers we have defined for our test scenario. Notice that the EIM Identifier can be any value and is not case sensitive.

Table 1-3 EIM Identifier

System i5 user	EIM Identifier	LDAP Registry	Registry type	Association type	User
ALTHOFF	Ursula Althoff	AS270DD.Due dorf.de.ibm.co m	OS/400	Target	ALTHOFF
		WebSphereRe gistry	1.3.18.0.2.33. 14-caselnor	Source	AS270DD.Due dorf.de.ibm.co m:389/URSUL A ALTHOFF
		WebSphereRe gistry	1.3.18.0.2.33. 14-caselnor	Source	URSULA ALTHOFF
JOHND	John Day	AS270DD.Due dorf.de.ibm.co m	OS/400	Target	JOHND
		WebSphereRe gistry	1.3.18.0.2.33. 14-caselnor	Source	AS270DD.Due dorf.de.ibm.co m:389/JOHN DAY
		WebSphereRe gistry	1.3.18.0.2.33. 14-caselnor	Source	JOHN DAY
WERKMANN	Dieter Werkmann	AS270DD.Due dorf.de.ibm.co m	OS/400	Target	WERKMANN
		WebSphereRe gistry	1.3.18.0.2.33. 14-caselnor	Source	AS270DD.Due dorf.de.ibm.co m:389/DIETE R WERKMANN
		WebSphereRe gistry	1.3.18.0.2.33. 14-caselnor	Source	DIETER WERKMANN
WFADV	Demo wfadv	AS270DD.Due dorf.de.ibm.co m	OS/400	Target	WFADV

System i5 user	EIM Identifier	LDAP Registry	Registry type	Association type	User
		WebSphereRegistry	1.3.18.0.2.33.14-caselnor	Source	AS270DD.Due dorf.de.ibm.com:389/DEMO WFADV
		WebSphereRegistry	1.3.18.0.2.33.14-caselnor	Source	DEMO WFADV
WFORG	Demo Wforg	AS270DD.Due dorf.de.ibm.com	OS/400	Target	WFORG
		WebSphereRegistry	1.3.18.0.2.33.14-caselnor	Source	AS270DD.Due dorf.de.ibm.com:389/DEMO WFORG
		WebSphereRegistry	1.3.18.0.2.33.14-caselnor	Source	DEMO WFORG

Enterprise Identity Mapping Configuration

Enterprise Identity Mapping (EIM) is a mechanism for mapping, or associating, a person or entity to the appropriate user identities in various registries throughout the enterprise.

The EIM Configuration wizard allows you to complete a basic Enterprise Identity Mapping configuration for your System i5 quickly and easily. The wizard provides you with three EIM system configuration options. How you use the wizard to configure EIM on a specific system depends on your overall plan for using EIM in your enterprise and your EIM configuration needs.

For example, many administrators want to use EIM in conjunction with Network Authentication Service (Kerberos) to create a single sign-on environment across multiple systems and platforms without a need to change underlying security policies. Consequently, the EIM Configuration wizard allows you to configure the Network Authentication Service as part of your EIM configuration. However, configuring and using Network Authentication Service is not a prerequisite or requirement for configuring and using EIM.

Once your planning is complete, you can use the EIM Configuration wizard to create one of the three basic EIM configurations. You can use the wizard to join an existing domain or to create and join a new domain. When you use the EIM

Configuration wizard to create and join a new domain, you can choose whether to configure an EIM domain controller on a local or a remote system.

2.1 Use the EIM Configuration wizard

We describe in the following steps how to use the EIM Configuration wizard when you start from scratch, which means we will create a new Directory server (LDAP).

To perform these tasks, you should install the iSeries Navigator on a client PC. The following tasks use the iSeries Navigator, which is packaged with iSeries Access for Windows®, which can be installed from your System i5. See “Installing iSeries Navigator” in the iSeries Information Center for details on installing iSeries Navigator. Ensure that you install all of the networking components, including TCP/IP. To access the iSeries Information Center, go to:

<http://publib.boulder.ibm.com/iseries/>

To configure EIM, you have to work with a user profile that has all of the following special authorities:

- ▶ Security administrator (*SECADM)
- ▶ All object (*ALLOBJ)
- ▶ System configuration (*IOSYSCFG)

To access the EIM Configuration wizard, follow these steps:

1. Start the iSeries Navigator.
2. Sign on to the System i5 for which you want to configure EIM. If you are configuring EIM for more than one System i5, begin with the one on which you want to configure the domain controller for EIM.
3. Expand **Network** → **Enterprise Identity Mapping**.
4. Right-click **Configuration** and select **Configure...** or **Reconfigure....** to launch the EIM Configuration wizard.

Note: After you have configured once, this option will be Reconfigure

Click **Help**, if necessary, to determine what information to specify as you proceed through the wizard.

5. Select **Create and join a new domain**, which creates a new EIM domain and allows you to configure a directory server as the EIM domain controller, and configures your system to participate in this new domain. Click **Next**.

Figure 2-1 shows the Welcome page of the EIM configuration wizard.

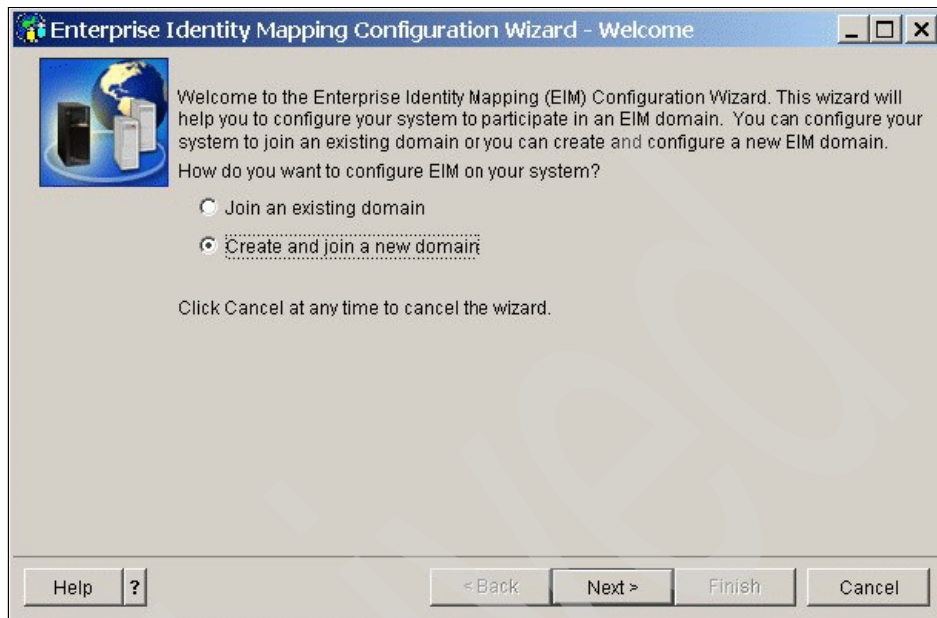


Figure 2-1 EIM Configuration Wizard - Welcome

6. Select **On the local Directory server** to configure the directory server on this system to act as the EIM domain controller (Figure 2-2 on page 33). Click **Next**.

Note: If Network Authentication Service is not currently configured on the System i5 the Network Authentication Services Configuration page displays. This page allows you start the Network Authentication Service Configuration wizard so that you can configure it. You can also configure Network Authentication Service at a later time by using the configuration wizard for this service through iSeries Navigator. After completing the network authentication service configuration, the EIM Configuration wizard continues.

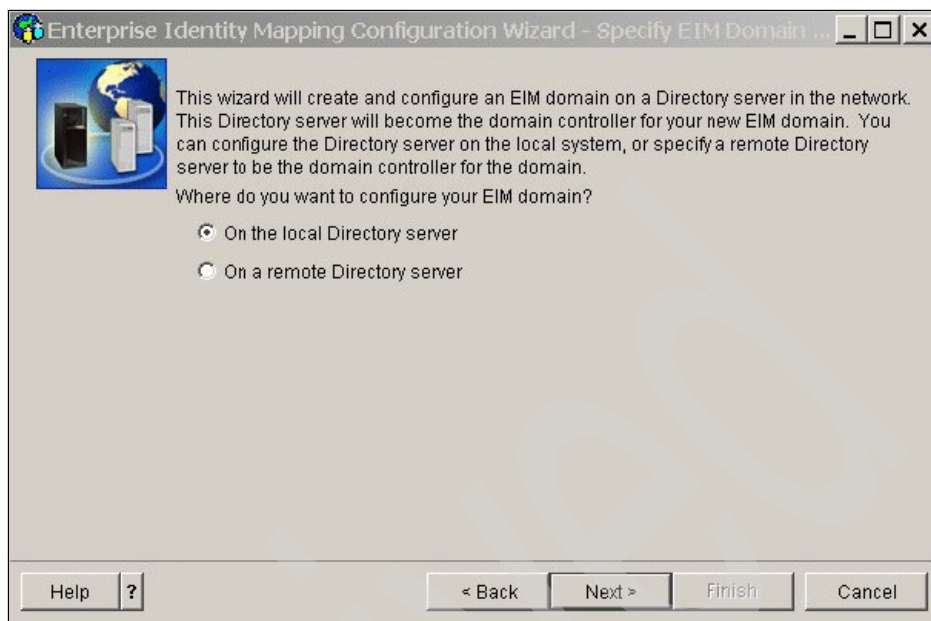


Figure 2-2 EIM Configuration Wizard - Specify EIM Domain

7. To configure the Network Authentication Service, select **Yes**, or you can choose to configure it later by selecting **No** (Figure 2-3 on page 34).

Note: On the Configure Network Authentication Service page, select **Yes** to start the Network Authentication Service Configuration wizard. With this wizard, you can configure several OS/400 interfaces and services to participate in a Kerberos realm as well as configure a single sign-on environment that uses both EIM and Network Authentication Service.

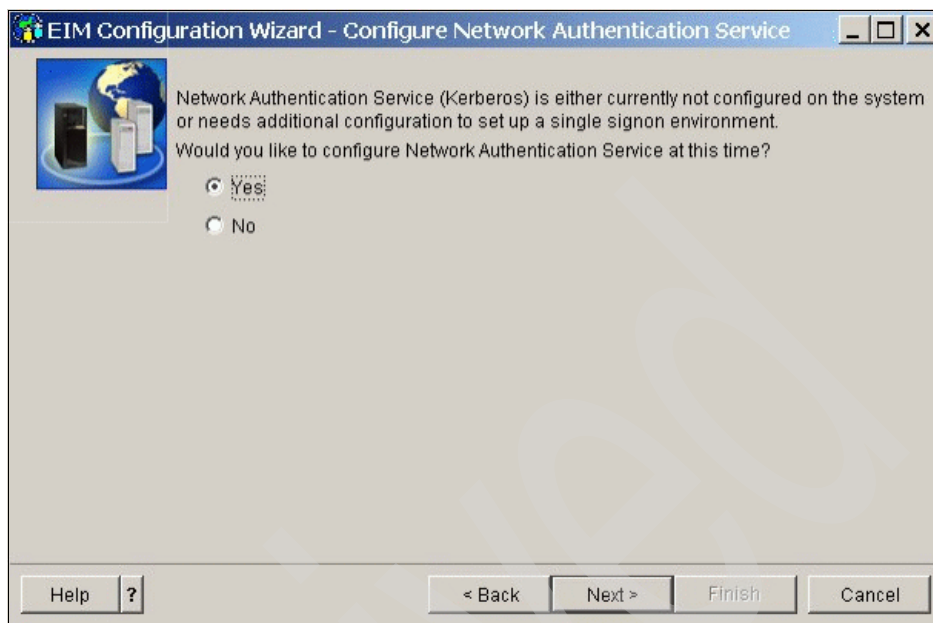


Figure 2-3 EIM Configuration Wizard - Configure Network Authentication Service

8. If the local directory server is not currently configured, the Configure Directory Server page displays when the EIM Configuration wizard resumes. Provide the following information to configure the local directory server:
 - a. Domain controller name. We use AS270DD.DUEDORF.DE.IBM.COM.
The name you use here will also become the directory name.
 - b. In the Port field, accept the default port number 389, or specify a different port number to use for non-secure EIM communications with the directory server. Click **Next**.

Note: If you configure the local directory server before you use the EIM Configuration wizard, the Specify User for Connection page displays instead. Use this page to specify the distinguished name and password for the LDAP administrator to ensure that the wizard has enough authority to administer the EIM domain and the objects in it and continue with the next step in this procedure.

Figure 2-4 shows the EIM domain controller page of the EIM Configuration Wizard.

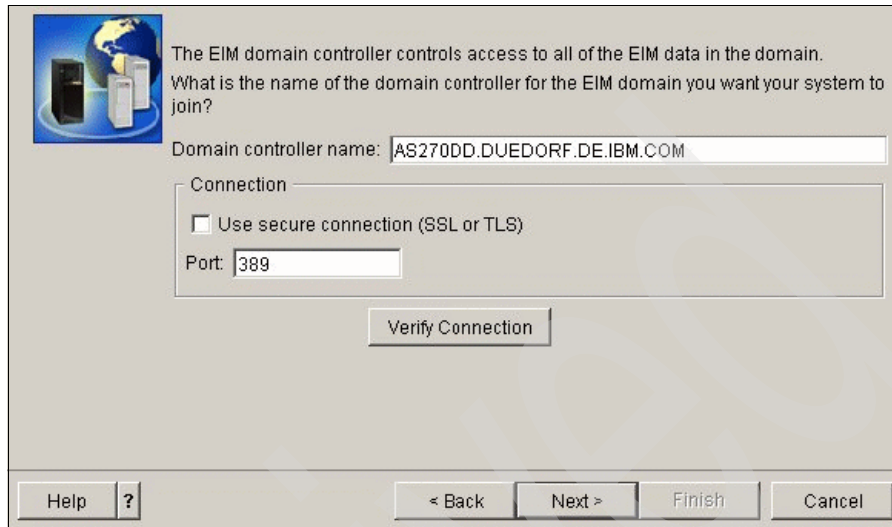


Figure 2-4 EIM Configuration Wizard - EIM domain controller

9. In the Distinguished name field, specify the LDAP distinguished name (DN) that identifies the LDAP administrator for the directory server (Figure 2-5 on page 36). The EIM Configuration wizard creates this LDAP administrator DN and uses it to configure the directory server as the domain controller for the new domain that you are creating.
 - In the Password field, specify the password for the LDAP administrator.
 - In the Confirm password field, specify the password a second time for validation purposes.
 - Click **Next**.

In order for the wizard to complete EIM configuration, the wizard must connect to the domain controller with an authorized user. What user do you want the EIM Configuration Wizard to use?

User type:

User

Distinguished name:

Password:

Confirm password:

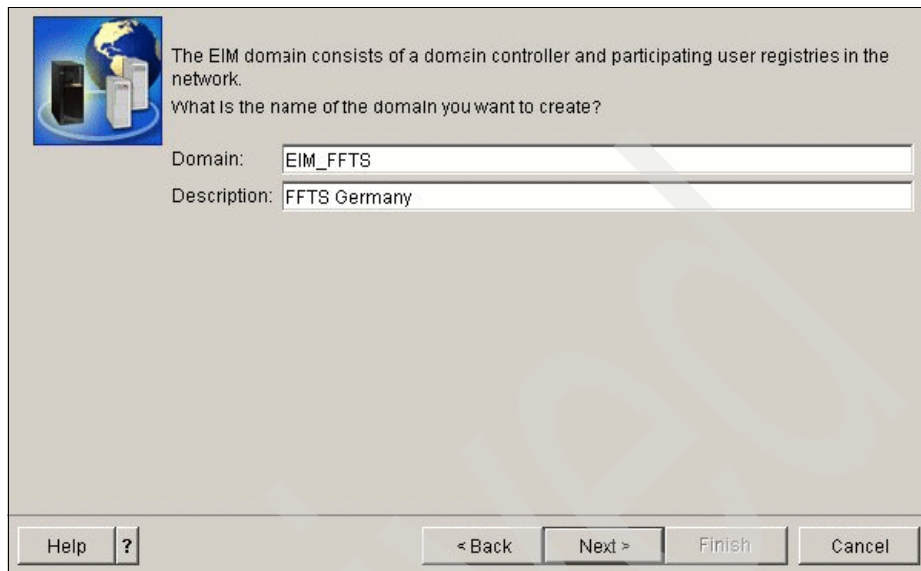
Figure 2-5 EIM Configuration Wizard - Define the LDAP administrator

10. On the Specify Domain page, provide the following information:

In the Domain field, specify the name of the EIM domain (we used EIM_FFIS) that you want to create. Accept the default name of EIM, or use any string of characters that makes sense to you. However, you cannot use special characters such as = + < > , # ; \ and *.

In the Description field, enter text to describe the domain. Click **Next**.

Figure 2-6 shows the Specify Domain page of the EIM Configuration Wizard.



The EIM domain consists of a domain controller and participating user registries in the network.
What is the name of the domain you want to create?

Domain:

Description:

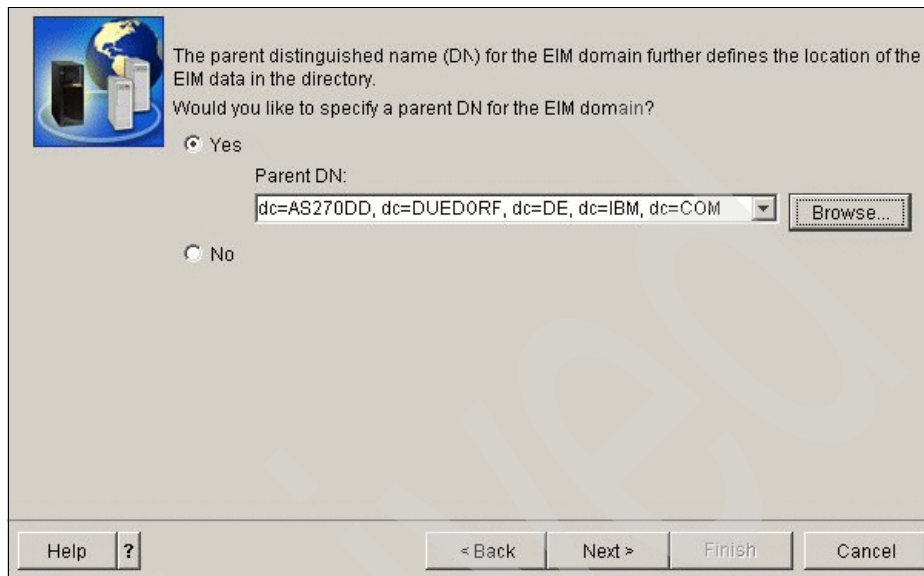
Help ? < Back Next > Finish Cancel

Figure 2-6 EIM Configuration Wizard - Specify Domain

11. On the Specify Parent DN for Domain page, select **Yes** to specify a parent DN for the domain that you are creating, or specify **No** to have EIM data stored in a directory location with a suffix whose name is derived from the EIM domain name.

Note: When you create a domain on a local directory server, a parent DN is optional. By specifying a parent DN, you can specify where in the local LDAP namespace EIM data should reside for the domain. When you do not specify a parent DN, EIM data resides in its own suffix in the namespace. If you select **Yes**, use the list box to select the local LDAP suffix to use as the parent DN, or enter text to create and name a new parent DN. It is not necessary to specify a parent DN for the new domain.

Figure 2-7 shows the Specify Parent DN for Domain of the EIM Configuration Wizard.



The screenshot shows a dialog box titled "Specify Parent DN for Domain". It contains a text box for "Parent DN:" with the value "dc=AS270DD, dc=DUED0RF, dc=DE, dc=IBM, dc=COM" and a "Browse..." button. There are radio buttons for "Yes" (selected) and "No". The dialog box also includes a "Help" button and a "Next >" button.

Figure 2-7 EIM Configuration Wizard - Specify Parent DN for Domain

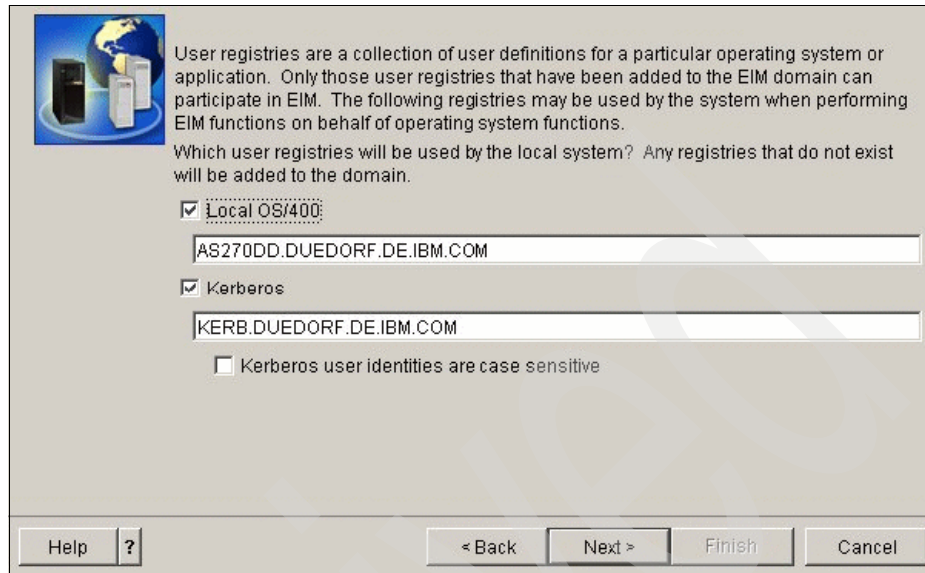
12. On the Registry Information page, shown in Figure 2-8 on page 39, specify whether to add the local user registries to the EIM domain as registry definitions. Select one or both of these user registry types.

Note: You do not have to create the registry definitions at this time. If you choose to create the registry definitions later, you need to add the system registry definitions and update the EIM configuration properties.

Select **Local OS/400** to add a registry definition for the local registry. In the field provided, accept the default value for the registry definition name or specify a different value for the registry definition name. The EIM registry name is an arbitrary string that represents the registry type and specific instance of that registry.

Select **Kerberos** to add a registry definition for a Kerberos registry. In the field provided, accept the default value for the registry definition name or specify a different value for the registry definition name. The default registry definition name is the same as the realm name. By accepting the default name and using the same Kerberos registry name as the realm name, you can increase performance in retrieving information from the registry. Select **Kerberos user identities are case sensitive**, if necessary.

Click **Next**.



The dialog box titled "EIM Configuration Wizard - Registry Information" contains the following elements:

- Icon:** A globe with a server tower and a mobile phone.
- Text:** "User registries are a collection of user definitions for a particular operating system or application. Only those user registries that have been added to the EIM domain can participate in EIM. The following registries may be used by the system when performing EIM functions on behalf of operating system functions."
- Text:** "Which user registries will be used by the local system? Any registries that do not exist will be added to the domain."
- Checkboxes:**
 - ☒ Local OS/400:
 - ☒ Kerberos
 - ☐ Kerberos user identities are case sensitive
- Text Fields:**
 - Below "Local OS/400": AS270DD.DUEDORF.DE.IBM.COM
 - Below "Kerberos": KERB.DUEDORF.DE.IBM.COM
- Buttons:** Help, ?, < Back, Next >, Finish, Cancel.

Figure 2-8 EIM Configuration Wizard - Registry Information

13. On the Specify EIM System User page, Figure 2-9 on page 41, select a User type that you want the system to use when performing EIM operations on behalf of operating system functions. These operations include mapping lookup operations and deletion of associations when deleting a local OS/400 user profile.

You can select one of the following types of users: **Distinguished name and password**, **Kerberos keytab file and principal**, or **Kerberos principal and password**. Which user types you can select vary based on the current system configuration. For example, if Network Authentication Service is not configured for the system, then Kerberos user types may not be available for selection. The user type that you select determines the other information that you must provide to complete the page as follows:

Note: You must specify a user that is currently defined in the directory server that is hosting the EIM domain controller. The user that you specify must have privileges to perform mapping lookup and registry administration for the local user registry at a minimum. If the user that you specify does not have these privileges, then certain operating system functions related to the use of single sign-on and the deletion of user profiles may fail. If you have not configured the directory server prior to running this wizard, the only user type you can select is Distinguished name and password and the only distinguished name you can specify is the LDAP administrator's DN.

- If you select **Distinguished name and password**, provide the following information:
 - In the Distinguished name field, specify the LDAP distinguished name that identifies the user for the system to use when performing EIM operations.
 - In the Password field, specify the password for the distinguished name.
 - In the Confirm password field, specify the password a second time for verification purposes.
- If you select **Kerberos principal and password**, provide the following information:
 - In the Principal field, specify the Kerberos principal name for the system to use when performing EIM operations.
 - In the Realm field, specify the fully qualified Kerberos realm name for which the principal is a member. The name of the principal and realm uniquely identifies the Kerberos users in the keytab file.
 - In the Password field, enter the password for the user.
 - In the Confirm password field, specify the password a second time for verification purposes.

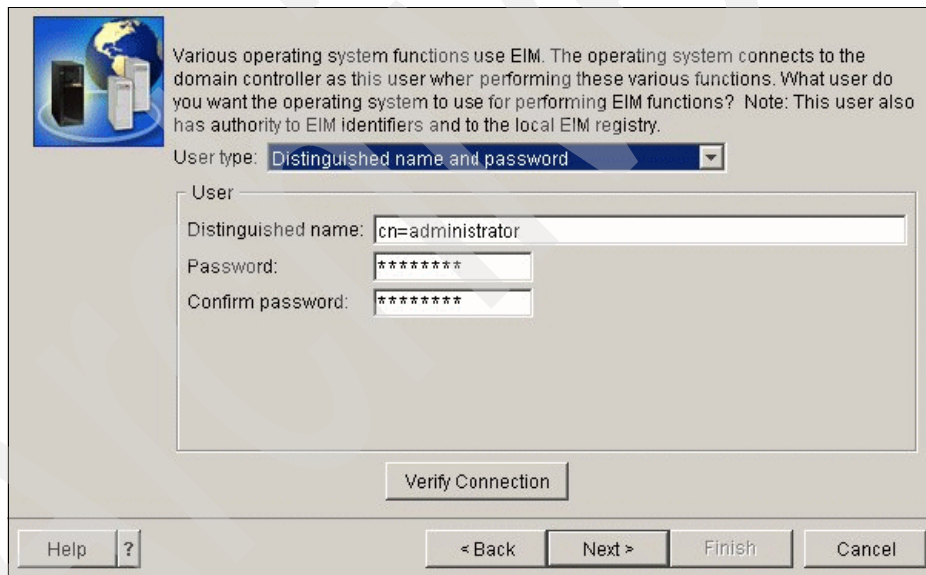
- If you select **Kerberos keytab file and principal**, provide the following information:

In the Keytab file field, specify the fully qualified path and keytab file name that contains the Kerberos principal for the system to use when performing EIM operations. Or click **Browse...** to browse through directories in the System i5 integrated file system to select a keytab file.

In the Principal field, specify the Kerberos principal name for the system to use when performing EIM operations.

In the Realm field, specify the fully qualified Kerberos realm name for which the principal is a member. The name of the principal and realm uniquely identifies the Kerberos users in the keytab file.

- Click **Verify Connection** to ensure that the wizard can use the specified user information to successfully establish a connection to the EIM domain controller.
- Click **Next**.



Various operating system functions use EIM. The operating system connects to the domain controller as this user when performing these various functions. What user do you want the operating system to use for performing EIM functions? Note: This user also has authority to EIM identifiers and to the local EIM registry.

User type: **Distinguished name and password**

User

Distinguished name: **cn=administrator**

Password: *********

Confirm password: *********

Verify Connection

Help **?** **< Back** **Next >** **Finish** **Cancel**

Figure 2-9 EIM Configuration Wizard - Specify EIM System User

14. In the Summary window, review the configuration information that you have provided. If all the information is correct, click **Finish**.

Figure 2-10 shows the EIM Domain Summary page of the EIM Configuration Wizard.

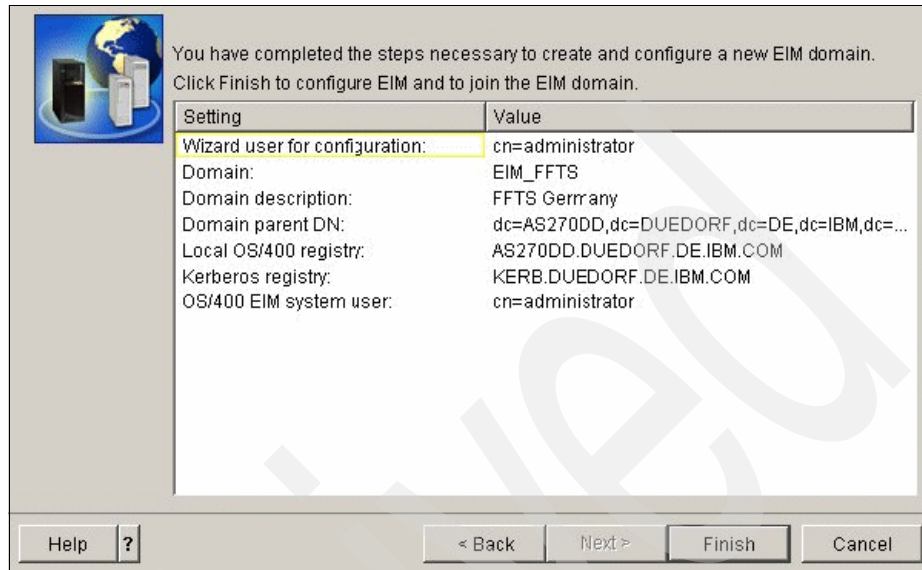


Figure 2-10 EIM Configuration Wizard - EIM Domain Summary

15. You will see a similar window showing the status of the configuration.

Figure 2-11 shows the EIM Configuration in process.

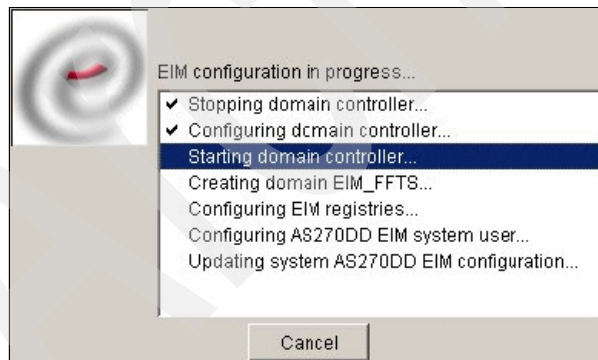


Figure 2-11 EIM Configuration in progress

16. When the wizard finishes, it adds the new domain to the Domain Management folder and a basic EIM configuration is created for this server.

In the iSeries Navigator you will see the new created EIM domain, as shown in Figure 2-12 on page 43.

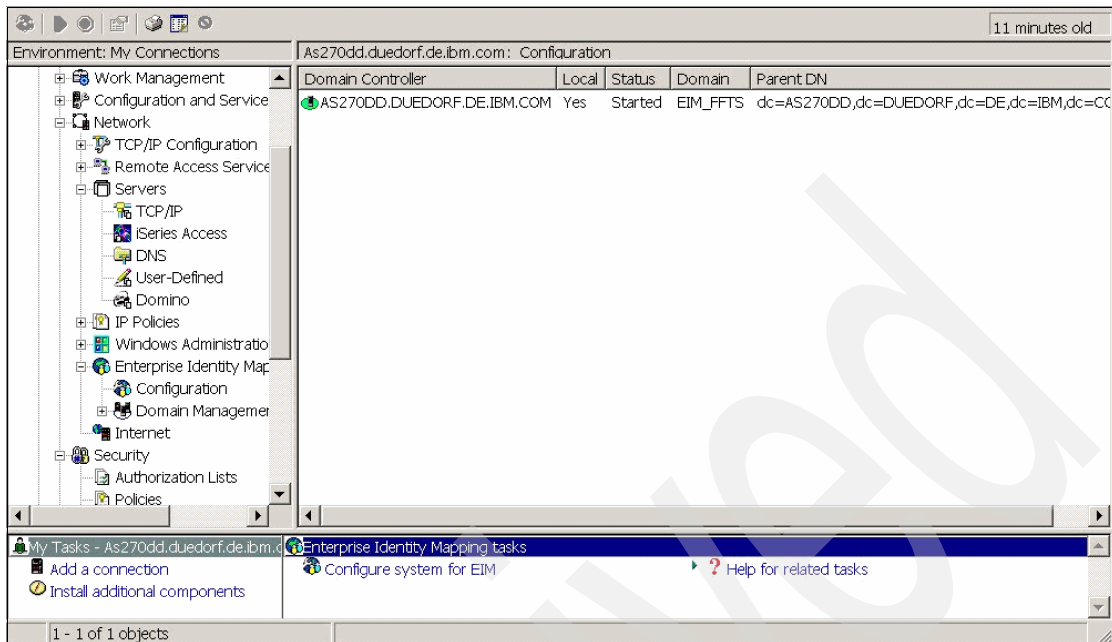


Figure 2-12 iSeries Navigator - New EIM Domain

2.2 Post configuration tasks

You must complete the following tasks to finalize your EIM configuration for the domain:

1. Use the EIM Configuration wizard on each additional server that you want to join the domain.
2. Add EIM registry definitions to the EIM domain, if necessary, for other non-System i5 servers and applications that you want to participate in the EIM domain. These registry definitions refer to the actual user registries that must participate in the domain. You can either add system registry definitions or add application registry definitions depending on your EIM implementation needs.
3. Based on your EIM implementation needs, determine whether to:
 - Create EIM identifiers for each unique user or entity in the domain and create identifier associations for them.
 - Create policy associations to map a group of users to a single target user identity.
 - Create a combination of these.

- Use the EIM test a mapping function to test the identity mappings for your EIM configuration.
- If the only EIM user you have defined is the DN for the LDAP administrator, then your EIM user has a high level of authority to all data on the directory server.

Therefore, you might consider creating one or more DNs as additional users that have more appropriate and limited access control for EIM data. The number of additional EIM users that you define depends on your security policy's emphasis on the separation of security duties and responsibilities. Typically, you might create at least the two following types of DNs:

- A user that has EIM administrator access control. This EIM administrator DN provides the appropriate level of authority for an administrator who is responsible for managing the EIM domain. This EIM administrator DN could be used to connect to the domain controller when managing all aspects of the EIM domain by means of iSeries Navigator.

At least one user that has all of the following access controls:

- Identifier administrator
- Registry administrator
- EIM mapping operations

This user provides the appropriate level of access control required for the system user that performs EIM operations on behalf of the operating system.

Note: For more information about EIM, see the iSeries Information Center at:

<http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp>
<http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp>

Configuring LDAP

A Lightweight Directory Access Protocol (LDAP) server is available as part of i5/OS in the product Directory Services for OS/400. The server provides a network directory that can be accessed by network clients using the LDAP protocol. LDAP defines the transport and format of messages used by a client to access data in an X.500-like directory. Although LDAP does not define the directory service itself, a directory accessed using LDAP is typically called an LDAP directory.

The directory server allows access to a type of database that stores information in a hierarchical structure similar to the way that the OS/400 integrated file system is organized. The LDAP directory server model is based on entries that consist of one or more attributes, such as a name or address, and a type. These attributes typically consist of mnemonic strings, such as cn for common name or mail for e-mail address.

Note: For more details on working with LDAP, refer to the iSeries Information Center under **Networking** → **TCP/IP** → **Directory Services** (LDAP).

When the EIM Configuration wizard is finished, see 2.1, “Use the EIM Configuration wizard” on page 31; your Directory Server has a basic configuration.

In this chapter, you find the next steps for configuring a LDAP directory.

You may want to do some or all of the following before continuing:

- ▶ Import data to the server; see Import an LDIF file in the iSeries Information Center.
- ▶ Enable Secure Sockets Layer (SSL) security; see Enable SSL on the Directory Server in the iSeries Information Center.

The link for the iSeries Information Center is:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/index.jsp>

or for i5/OS v5R4

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/index.jsp>

- ▶ We already configured a Directory Server (with the EIM Configuration wizard; see 2.1, “Use the EIM Configuration wizard” on page 31) and now we create a directory database that will contain employee information such as names, e-mail addresses, telephone numbers, and so on.
- ▶ All employees (managers and non-managers) will exist in the employees directory tree. Managers also belong to the managers group. Members of the managers group have authority to change employee data.
- ▶ We can use the Directory Server Web Admin Tool to make our definitions for the directory server.
- ▶ Before we can use the Directory Server Web Admin Tool we have to set it up. See “Set up the Directory Server Web Administration Tool” on page 47. Then do the basic configuration for the Web Admin Tool, as described in “Configure the Directory Server Web Administration Tool” on page 48.
- ▶ Next we create the directory database as described in 3.2, “Create the directory database” on page 51.
- ▶ Beginning with 3.3, “Templates and realms” on page 53, we make our definitions to add employee information into our directory database.
- ▶ Because we have a huge number of System i5 users for which the system distribution directory (SDD) also exists, we publish the data residing in the SDD to the LDAP directory; see 3.4, “Publish SDD data to the directory database” on page 60.

3.1 Directory Server Web Administration tool

One or more Directory Servers can be administered through the Directory Server Web Administration Tool. This Web administration console allows you to:

- ▶ Add or change the list of Directory Servers that can be administered
- ▶ Change Web administration console attributes
- ▶ Administer a Directory Server

Set up the Directory Server Web Administration Tool

Perform the following steps to set up the Directory Server Web Administration Tool for the first time.

1. Install IBM WebSphere Application Server V6.0 (5733-W60 Base or Express options) and the associated prerequisite software if they are not already installed.
2. Start the HTTP ADMIN server instance by doing one of the following:
 - a. In iSeries Navigator, select **Network** → **Servers** → **TCP/IP** and right-click **HTTP Administration**. Then click **Start**.
Or on a command line type:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```
3. Log in to the IBM Web Administration for iSeries. Use an operating system user profile and password to log in to the iSeries Tasks page, which you open through your browser with `http://your_server:2001`, then click **IBM Web Administration for iSeries**.
4. From the HTTP Server Administration page, click the **Manage** tab and then click the **HTTP Servers** tab. Make sure **ADMIN - Apache** is selected in the Server drop-down list and that **Include /QIBM/UserData/HTTPAdmin/conf/admin-cust.conf** is selected in the Server Area drop-down list.
5. From the options in the left pane of the page, click **General Server Configuration**.

Note: You might need to expand the section **Server Properties** in order to see the General Server Configuration option.

6. Set Start the system application server instance when the 'Admin' server is started to **Yes**.
7. Click **OK**.

8. Restart the HTTP ADMIN server instance by clicking the **Restart** button. You can also stop and start the HTTP ADMIN server instance using the iSeries Navigator or a command line.

- In the iSeries Navigator, select **Network** → **Servers** → **TCP/IP** and right-click **HTTP Administration**. Then click **Stop**.

OR

- On a command line type:

```
ENDTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

Start the HTTP ADMIN server instance again as described in step number 2.

Configure the Directory Server Web Administration Tool

1. Log in to the Directory Server Web Administration Tool. Bring up the Login page by doing one of the following:

- From iSeries Navigator, select your server and select **Network** → **Servers** → **TCP/IP**, right-click **IBM Directory Server**, and click **Server Administration**.

OR

- From the iSeries Tasks page (http://your_server:2001), click **IBM Directory Server for iSeries**.

2. A Login page should appear. (Figure 3-1) In the LDAP Hostname list, select **Console Admin**. Type superadmin for the username and secret for the password (default values). Click **Login**.

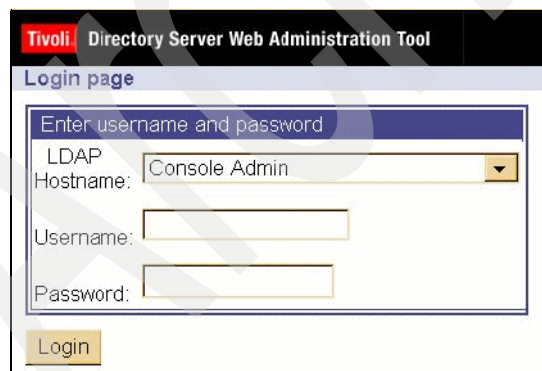


Figure 3-1 Directory Server Web Administration Tool - Login page

3. Expand **Console administration** and click the link **Change console administrator login**.

Figure 3-2 shows the Console administration of the Directory Server Web Administration Tool.



Figure 3-2 Directory Server Web Administration Tool - Console administration

4. In this window (Figure 3-3), change the default name of the console administrator from superadmin to a name you want. For this scenario, we use *administrator*. Click **OK**.

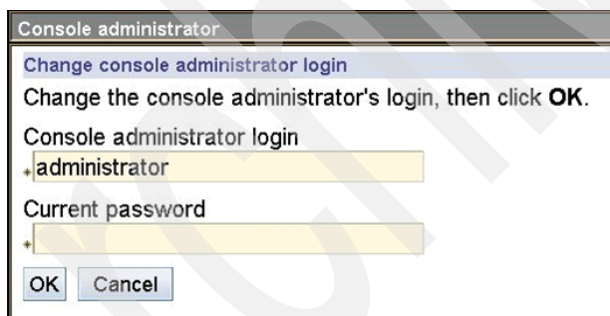
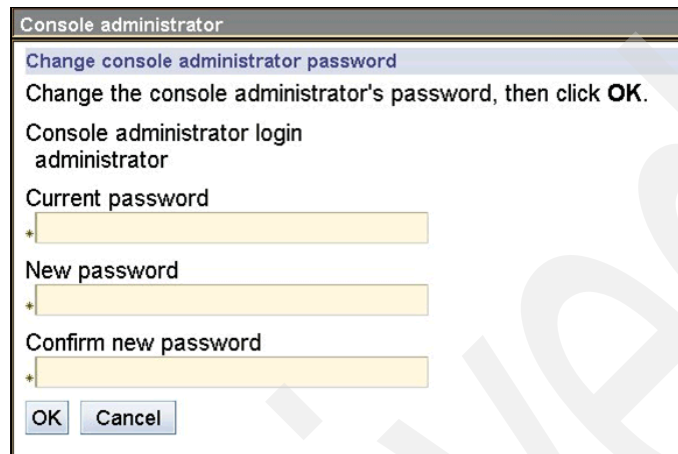


Figure 3-3 Directory Server Web Administration Tool - Change console administrator login

5. Back in the console administrator window (Figure 3-2 on page 49), click the link **Change console administrator password**. In the window that comes up (Figure 3-4), change the **password** for the console administrator by supplying the current password, the new password, confirm, and then click **OK**.



Console administrator

Change console administrator password

Change the console administrator's password, then click **OK**.

Console administrator login
administrator

Current password

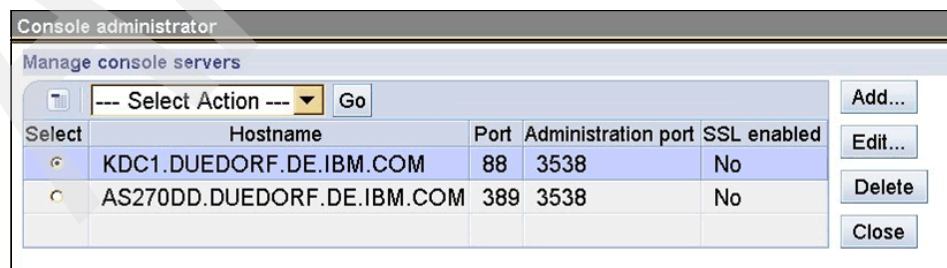
New password

Confirm new password

OK **Cancel**

Figure 3-4 Directory Server Web Administration Tool - Change console administrator password

6. Configure the Web Administration tool to connect to the LDAP server on your System i5. Select **Console administration** → **Manage console servers** in the left hand navigation. This is shown in Figure 3-5.
 - a. Click **Add**.
 - b. In the Add server field, type AS270DDDUEODRF.DE.IBM.COM; remember that this is the name of the directory server we configured in Chapter 2, “Enterprise Identity Mapping Configuration” on page 29.
 - c. Click **OK**. The new server appears in the list under Manage console servers. See Figure 3-5.



Console administrator

Manage console servers

--- Select Action --- Go

Select	Hostname	Port	Administration port	SSL enabled
<input type="radio"/>	KDC1.DUEDORF.DE.IBM.COM	88	3538	No
<input type="radio"/>	AS270DD.DUEDORF.DE.IBM.COM	389	3538	No

Add... Edit... Delete Close

Figure 3-5 Directory Server Web Administration Tool- Manage console servers

- d. Click **Logout** in the left hand navigation.
7. Open the Web administration tool again. In the LDAP Hostname list, select the server you just added (**AS270DD.DUEDORF.DE.IBM.COM**).

In the Username field, type cn=administrator (or the name you enter in step 4 on page 49), and in the Password field type the according value. You should see the main page of the IBM Directory Server Web Administration tool.

3.2 Create the directory database

From the Directory Server Web Administration Tool, we create a new directory using the following steps:

1. Select **Directory management** → **Manage entries**.
You see a listing of the objects in the base level of the directory. Since the server is new, you see only the structural objects that contain the configuration information.
2. You want to add a new object to contain the directory data. First click **Add...** on the right side of the window.
3. In the next window, scroll within the Object class list to select **domain** and click **Next**. Figure 3-6 shows how to add an entry in the Directory Server Web Administration Tool.

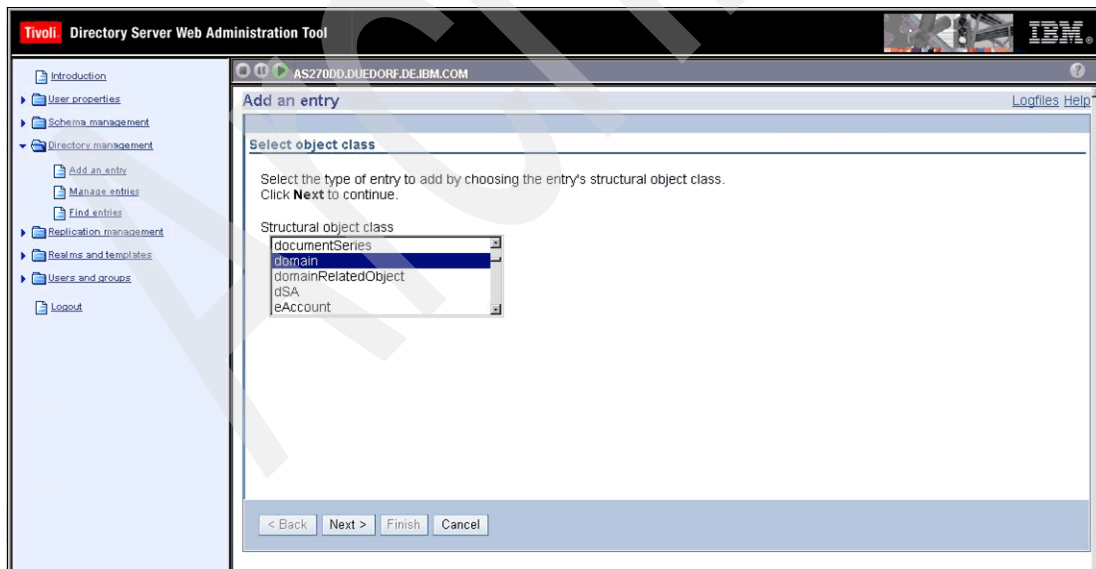


Figure 3-6 Directory Server Web Administration Tool - Directory Management - Add an entry

4. You do not want to add any auxiliary object classes, so click **Next** again.
5. In the Enter the attributes window, enter the data that corresponds with the suffix that you created earlier in the wizard. Leave the Object class drop-down list on domain. Type (or select when already created) dc=as270dd,dc=duedorf,dc=de,dc=ibm,dc=com for the relative DN; the Parent DN field should be empty.
6. Click **Finish** at the bottom of the window. Back in the base level you should see the new base DN. Figure 3-7 shows the window provided to enter the attributes in the Directory Server Web Administration Tool.

AS270DD.DUEDORF.DE.IBM.COM

Enter the attributes

Enter the values for the attributes of the new entry. For multiple values click **Multiple values** next to the attribute.

When you have entered all the required attributes and any of the other attributes click **Finish** at the bottom of the page.

Object class
domain

Distinguished name (DN)

Relative DN: dc=as270dd,dc=du Parent DN:

Required attributes dc

[Other attributes](#)

< Back Next > Finish Cancel

Figure 3-7 Directory Server Web Administration Tool - Directory Management - Enter the attributes

3.3 Templates and realms

In this section we present a short overview about templates and realms in the directory server. For more information, go to the iSeries Information Center:

<http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html>

A template describes what a user looks like. It specifies the object classes that are used when creating users (both the structural object class and any auxiliary classes that you want). A template also specifies the layout of the windows used to create or edit users (for example, names of tabs, default values, and attributes to appear on each tab).

A realm identifies a collection of users and groups. It specifies information, in a flat directory structure, such as where users are located and where groups are located. A realm defines a location for users (for example, `?cn=employees,o=acme,c=us?`) and creates users as immediate subordinates of that entry (for example John Day is created as `?cn=John Day,cn=employees,o=acme,c=us?`). You can define multiple realms and give them familiar names (for example, Web Users). The familiar name can be used by the people that are creating and maintaining the users.

3.3.1 Create a user template

Create a user template as an aid to adding the employee data.

1. Point your browser to `http://as270dd.duedorf.de.ibm.com:2001` and provide the System i5 user ID and password, which opens the iSeries Task window.
2. Click the **IBM Directory Server for iSeries** link to open the Directory Server Web Administration Tool.
3. The Login page appears. In the LDAP Hostname list, select your LDAP Hostname, in our case, `AS270DD.DUEDORF.DE.IBM.COM`. In the field username, type the LDAP administrator name, in our case `cn=administrator`, and provide the password. Click **Login**.

- Click **Realms and templates** → **Add user template**, as shown in Figure 3-8.



Figure 3-8 Directory Server Web Administration Tool - Realms and templates

- In the User template name field, type **employee**. Click the **Browse...** button next to the Parent DN field.
- Select the base DN you created in the previous section by toggling on the radio button (in our case next to **dc=as270dd,dc=duedorf,dc=de,dc=ibm,dc=com**) and click **Select** in the right of the window. This is shown in Figure 3-9.

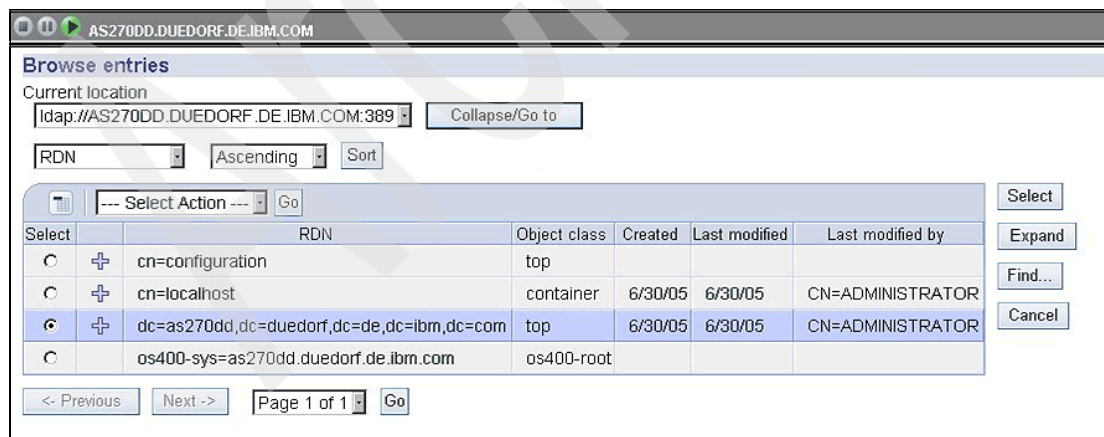


Figure 3-9 Directory Server Web Administration Tool - Browse entries

7. You are taken back to the Add user template (Figure 3-10). Click **Next**.
8. In the Structural object class drop-down list, choose **inetOrgPerson** and click **Next**, as shown in Figure 3-10.

AS270DD.DUEDORF.DE.IBM.COM

Add user template [Logfiles Help](#)

Template

Structural object class
inetOrgPerson

Auxiliary object classes

Available object classes

aixAuxAccount
aixAuxGroup
aliasObject
bootableDevice
cacheObject

Add >>
<< Remove

Selected obj
[Empty]

697x163

< Back Next > Finish Cancel

Figure 3-10 Directory Server Web Administration Tool - Add user template

9. In the Naming attribute drop-down list, select **cn**. In the Tabs list, select **Required** and click **Edit**, as shown in Figure 3-11.

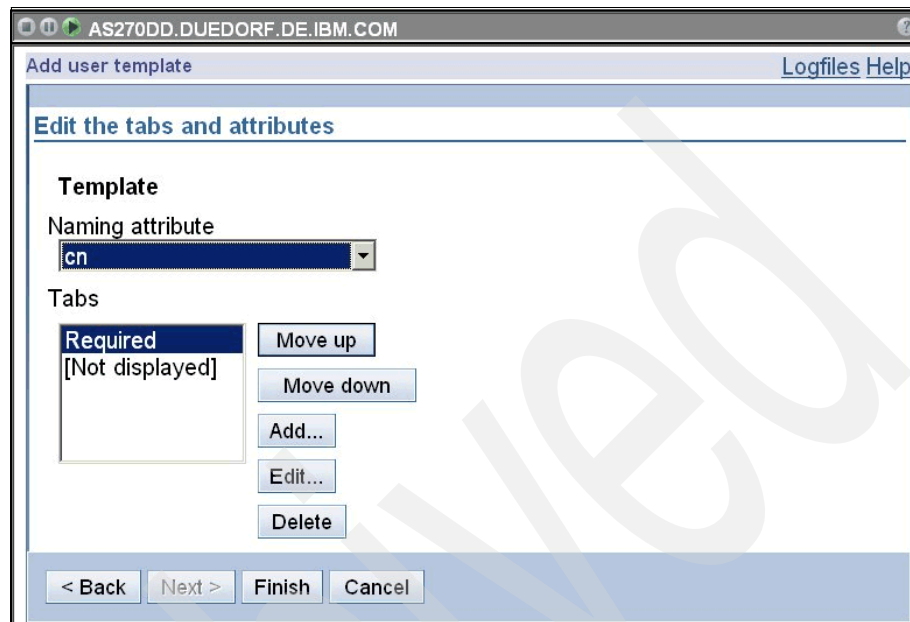


Figure 3-11 Directory Server Web Administration Tool - Add user template-attributes

10. In the Edit tab window (Figure 3-12 on page 57) choose which fields to include in the user template (sn and cn are required).
- In the Attributes list, select **departmentNumber** and click **Add >>**.
 - Select **telephoneNumber** and click **Add >>**.
 - Select **mail** and click **Add >>**.
 - Select **userPassword** and click **Add >>**.
 - Click **OK** and then **Finish**. The user template will be created.

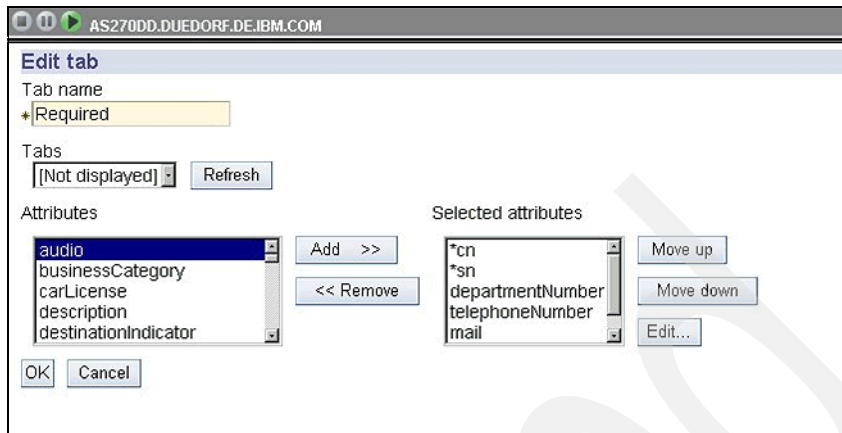


Figure 3-12 Directory Server Web Administration Tool - Add user template - assign attributes

3.3.2 Create a realm

1. In the Web Administration tool, select **Realms and templates** → **Add realm**.
2. In the Realm name field, type employees.
3. Click **Browse...** to the right of the Parent DN field.
4. Select the parent DN you created (ours is **dc=as270dd,dc=duedorf,dc=de,cd=ibm,dc=com**) and click **Select** in the right side of the window. Go back to the Add Realm window and click **Next**.

5. In the Add realm window (Figure 3-13) you only need to change the User template drop-down list. Select the user template you created (**cn=employee, dc=as270dd, dc=duedorf, dc=de, cd=ibm, dc=com**). Click **Finish**.

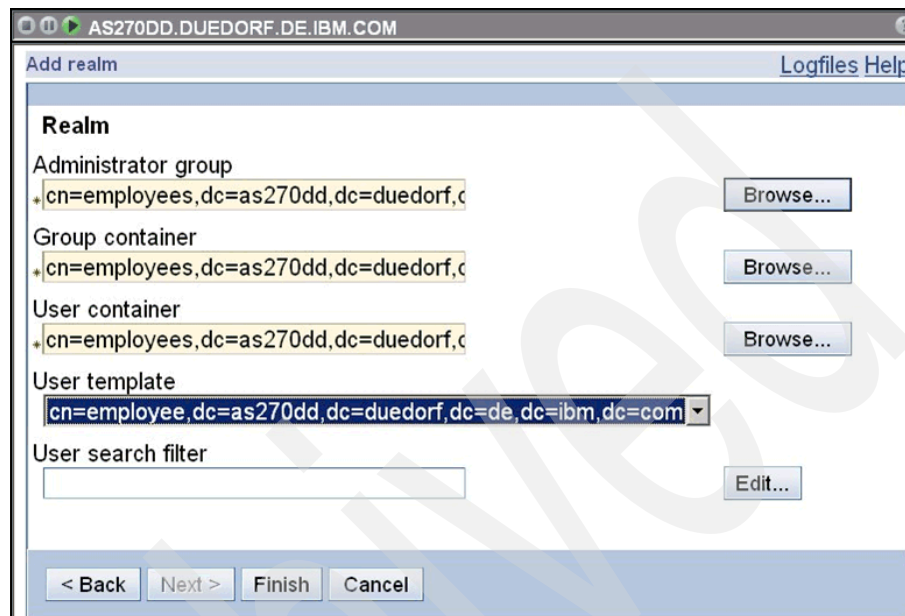


Figure 3-13 Directory Server Web Administration Tool - Directory Management - Add realm

3.3.3 Access control lists

Access control lists (ACLs) provide a means to protect information stored in a LDAP directory. Administrators use ACLs to restrict access to different portions of the directory, or specific directory entries.

It is best to design your access control strategy by creating groups of users that you will use when setting the access for objects and attributes. Set ownership and access at the highest level in the tree possible and let the controls cascade down the tree. Even though this step is not essential for SSO, we will provide the steps to implement an access control strategy.

Create a manager group

We create a group called managers that will be the owner of the ACL for our employees realm.

1. Select **Users and groups** → **Add group**.
2. In the Group name field, type managers.

3. Ensure that **employees** is selected in the Realm drop-down list.
4. Click **Finish**.

Configure the manager group as administrator

To configure the manager group as an administrator for the employees realm, do the following:

1. Select **Realms and templates** → **Manage realms**.
2. Select the realm that you created, **cn=employees, dc=as270dd,dc=duedorf,dc=de,cd=ibm,dc=com**, and click **Edit**.
3. To the right of the Administrator group field, click **Browse....**
4. Select **dc=as270dd,dc=duedorf,dc=de,cd=ibm,dc=com** and click **Expand**.
5. Select **cn=employees** and click **Expand**.
6. Select **cn=managers** and click **Select**.
7. In the Edit realm window, click **OK**.

Give the manager group authority

To give the manager group authority over the **c=as270dd,dc=duedorf,dc=de,cd=ibm,dc=com** suffix, do the following:

1. Select **Directory management** → **Manage entries**.
2. Select **dc=as270dd,dc=duedorf,dc=de,cd=ibm,dc=com** and click the **Edit ACL....** button.
3. In the next window, click the **Owners** link.
4. Select the **Propagate owner** check box (Figure 3-14 on page 60). Everyone who is a member of the managers group will be made an owner of the **dc=as270dd,dc=duedorf,dc=de,cd=ibm,dc=com** data tree.
5. In the Type drop-down list, select **group**.
6. In the DN (Distinguished name) field, select **cn=managers,cn=employees, dc=as270dd,dc=duedorf,dc=de,cd=ibm,dc=com**, see Figure 3-14 on page 60.
7. Click **Add**.
8. Click **OK**.

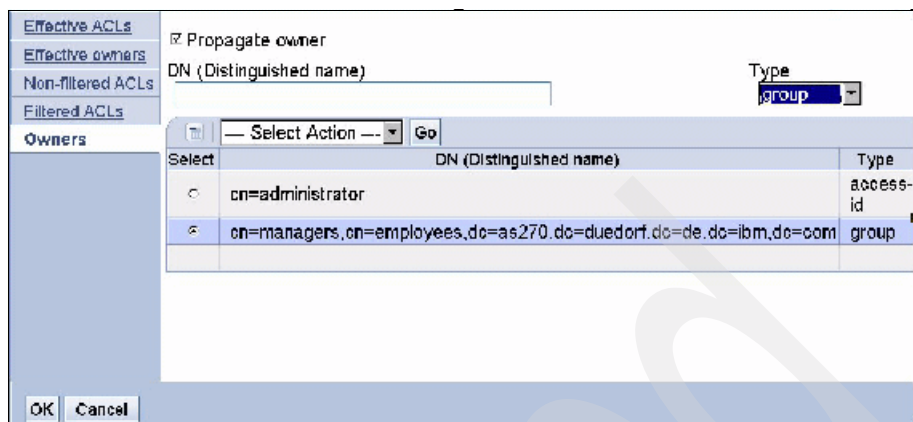


Figure 3-14 Directory Server Web Administration Tool - Directory Management - Edit ACL

Add a user to the manager group

1. In the Web Administration tool, click **Users and groups** → **Manage groups**.
2. Select the realm you created, *employees*, in the Realm menu, and be sure managers is selected under Groups. Click **Edit**.
3. Select the users that should be added to the manager group from the available users selection list and click the **ADD >>** button. Click **OK**.
4. Log out of the Web administration tool by clicking **Log out** in the left hand navigation.

3.4 Publish SDD date to the directory database

Because most of System i5 user profiles are also in the system distribution directory (SDD) we publish the data residing in the SDD to the LDAP directory.

Note: When you create i5/OS user profiles using the iSeries Navigator you have both a user profile and a System Distribution Directory user entry. If you use CL commands to create user profiles in your System i5 environment, you have to create a user profile (CRTUSRPRF) and a System Distribution Directory (SDD) user entry with the command Add Directory Entry (ADDDIRE). If your users exist only as a user profile and you want them to be published to the LDAP directory, you have to create system distribution directory user entries for them.

You can configure your system to publish certain information into a Directory Server on the same system or on a different system as well as user defined information. The operating system automatically publishes this information to the Directory Server when you use iSeries Navigator to define the publishing (see 3.4.1, “Setting up SDD publishing” on page 63). This also keeps the LDAP directory synchronized with changes that are made in the system distribution directory.

Published users can also be used to support LDAP authentication with some users published from the system distribution directory, and other users added to the directory by other means. A published user has a `uid` attribute that names the user profile, and has *no* `userPassword` attribute. When a bind request is received for an entry like this, the server calls the operating system security to validate the uid and password as a valid user profile and password for that profile. If you want to use LDAP authentication, and would like existing users to be able to authenticate using their operating system passwords, while non-i5/OS users are added to the directory manually, you should consider this function.

Note: The call to the operating system to validate the uid and password works only in the above mentioned way, if the Directory Server runs on System i5.

If the parent DN to which the data is being published does not exist, Directory Server automatically creates it. You might have also installed other i5/OS applications that publish information in an LDAP directory. Additionally, you can call application program interfaces (APIs) from your own programs to publish other types of information to the LDAP directory.

Note: You can also publish i5/OS information to a directory server that is not running on i5/OS if you configure that server to use the IBM schema.

The System Distribution Directory entry is exported to the LDAP directory by using the `inetOrgPerson` object class and the `ePerson` object class. Table 3-1 describes the mapping of System Distribution Directory fields to attributes of the `inetOrgPerson` and `ePerson` object class.

Table 3-1 Mapping of System Distribution Directory

System Distribution Directory field	LDAP attribute
User profile	uid
Description	description
Last name	sn (surname), cn (common name)
First name	givenName, cn (common name)

System Distribution Directory field	LDAP attribute
Preferred name	cn (common name)
Full name	cn (common name)
User ID	cn (common name)
Department	departmentNumber
Job title	title
Telephone number 1 and 2	Telephone number 1 and 2
FAX telephone number	fascimileTelephoneNumber
Office	roomNumber
Address lines 1-4	registeredAddress
SMTP name	mail

The common name (cn) will use the following formats:

- ▶ 'First name' 'Middle Name' 'Last name'
- ▶ 'Preferred name' 'Last name'
- ▶ 'Full name'
- ▶ 'UserID'

For example:

A user with the first name of Jonathan, preferred name of John, middle name of T, last name of Smith, and user ID of JSMITH, would have the following common names:

- ▶ cn=Jonathan T. Smith
- ▶ cn=John Smith
- ▶ cn=Smith, Jonathan T. (John)
- ▶ cn=JSMITH

The distinguished name (DN) of the published entry is the first common name (cn) combined with the directory path.

For example:

If the directory path is ou=employees, o=iseriesshop, the distinguished name (dn) for this user would be cn=Jonathan T. Smith,ou=employees,o=iseriesshop.

If you have two users in the System Distribution Directory that will resolve to the same DN, they will overlay each other in the LDAP server. Sometimes overlaying names is what you want if you are merging multiple OS/400 SDDs into one LDAP server. If you have different users with the same name, ensure they have different distinguished names to prevent overlaying each other.

Note: The best time to clean up name conflicts is before publishing the SDD for the first time.

3.4.1 Setting up SDD publishing

To configure the System i to publish data from the System Distribution Directory (SDD) into the LDAP directory, do the following:

1. In iSeries Navigator, right-click your System i5 in the left hand navigation and select **Properties**.
2. In the Properties dialog box, choose the **Directory Services** tab.
3. Select **Users** and click **Details**.
4. Select the **Publish user information** check box.
5. In the Where to publish section, click the **Edit** button.
6. In the next window, select or type in your server, in our case AS270DD.DUEDORF.DE.IBM.COM, and click **OK**.
7. In the Under DN field, use the **Browse** button to select the according DN (in our scenario it will be cn=employees,dc=AS270DD,dc=DUEDORF,dc=DE,dc=IBM,dc=COM) where employees is the name of the user template you created in 3.3.1, "Create a user template" on page 53.
8. In the Server connection section, ensure that the default port number, 389, is entered in the Port field. In the User type drop-down list, choose **Distinguished name** and enter cn=adminstrator in the Distinguished name field.
9. Click **Password**.

Figure 3-15 shows the Configuration tab on the System i5 to publish data.

The screenshot shows a window titled "User Information Details" with a "Configuration" tab. Inside, there's a section "Publish user information" with a checked checkbox. Below it, "Where to publish" shows a "Directory server" field with the value "AS270DD.DUEDORF.DE.IBM.COM" and an "Under DN" dropdown menu showing "cn=employees,dc=AS270DD,dc=DUEDORF,dc=DE,d". There are "Edit" and "Browse..." buttons. The "Server connection" section has an unchecked checkbox for "Use Secure Sockets Layer (SSL)", a "Port" field with "389", a "User type" dropdown menu with "Distinguished name", and a "Distinguished name" field with "cn=adminstrator". A "Password..." button is below this. At the bottom of the window are "Verify", "OK", "Cancel", "Help", and "?" buttons.

Figure 3-15 Configure the System i5 to publish data

10. Type the LDAP administrator password in the Password field and click **OK**.
11. Click the **Verify** button. This ensures that you have entered all the information correctly and that the System i5 can connect to the LDAP directory.
12. Click **OK**.

The publishing task

By default the publishing task will take place every five minutes. You can also of start the publishing task by calling the program QGLDSSDD. Here is a sample how to call the publishing task:

```
CALL PGM(QGLDSSDD) PARM(*CHG 'cn=adminstrator' 'secret' 0 0 0)
```

where the administrator and the followed password have to be the names of your environment.

For more information about the publishing tasks jobs, see the System i Information Center or go to:

<http://publib.boulder.ibm.com/series/>

3.5 Create a user for the WebSphere Administrator

In this step, you create an LDAP directory entry for a user that will be used later as a user for the WebSphere Administrator.

1. Point your browser to `http://yourServer:2001` and provide the System i5 user ID and password, which opens the iSeries Task window.
2. Click link **IBM Directory Server for iSeries** link to open the Administrator in the IBM Directory Server.
3. The Login page appears. In the LDAP Hostname list, select your LDAP Hostname, in our case `AS270DD.DUEDORF.DE.IBM.COM`. In the username field, type the LDAP administrator name, in our case `cn=admin` and provide the password. Click **Login**.

Figure 3-16 shows the Login page of the Directory Server Web Administration Tool.

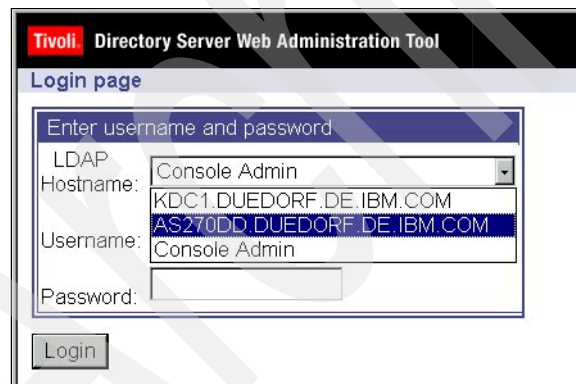


Figure 3-16 Login page of the Directory Server Web Administration Tool

4. Expand **User and groups**.

5. Click the link **Manage users**. See Figure 3-17.

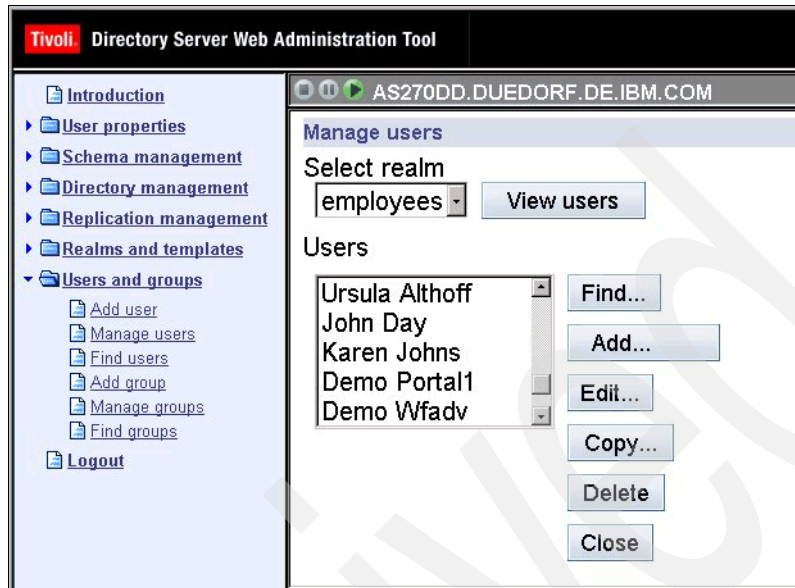


Figure 3-17 Directory Server Web Administration Tool - Manage users

6. Click the **Add** button.

7. Select the **employees** realm that you created in 3.3.2, “Create a realm” on page 57. Click **Next**. See Figure 3-18.

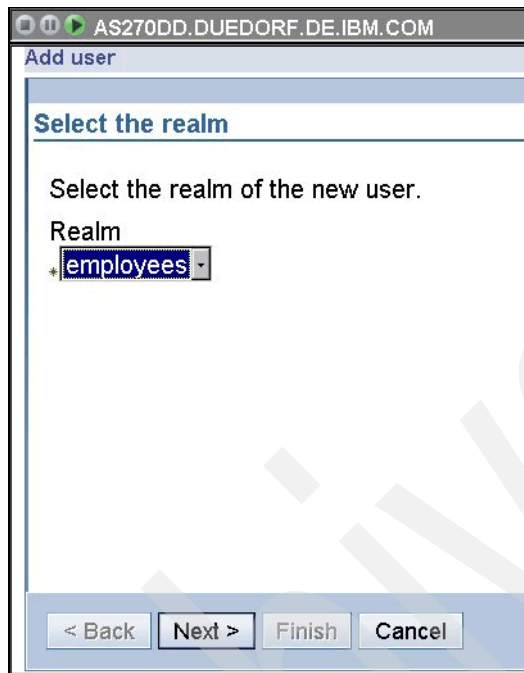


Figure 3-18 Directory Server Web Administration Tool - Select the realm

8. Type the values for the naming attributes for the new LDAP user entry shown in Figure 3-19. Click **Finish**.

AS270DD.DUEDORF.DE.IBM.COM

Add user

Realm
employees

Naming attribute

cn
WASADMIN1

Required

User groups

*sn
WASADMIN1 Multiple values

*cn
WASADMIN1 Multiple values

departmentNumber
2005 Multiple values

telephoneNumber
123456 Multiple values

mail
wasadmin1@de.ibm.com Multiple values

userPassword
..... Multiple values

< Back Next > Finish Cancel

Figure 3-19 Directory Server Web Administration Tool - Add user naming attribute

3.6 Test the directory database

Search the directory database using the **ldapsearch** command line.

1. On the command-line interface, enter the command QSH to open a Qshell session.
2. Enter the following to retrieve a list of all the LDAP entries in the database:

```
ldapsearch -h as270dd.duedorf.de.ibm.com -b  
dc=as270dd,dc=duedorf,dc=de,dc=ibm,dc=com objectclass=*
```

(Type this command as one line.)

Where -h is the name of the host machine running the LDAP server, -b is the base DN to search under, and objectclass=* returns all of the entries in the directory.

This command returns information similar that as shown in Example 3-1.

Example 3-1 Result of the ldapsearch command

```
dc=AS270DD,dc=DUEODRF,dc=DE,dc=IBM,dc=COM
dc=AS270DD
objectclass=top
objectclass=domain
..
cn=Demo Wfadv,cn=employees,dc=AS270DD,dc=DUEODRF,dc=DE,dc=IBM,dc=COM
objectClass=top
objectClass=person
objectClass=organizationalPerson
objectClass=inetOrgPerson
objectClass=publisher
objectClass=ePerson
cn=Demo Wfadv
cn=Wfadv\, Demo
cn=WFADV
sn=Wfadv
uid=WFADV
givenName=Demo
description=Testusr for WAS SSO
departmentNumber=2005
registeredAddress=wfadv@de.ibm.com
mail=WFADV?I825D@AS270DD.DUEODRF.DE.IBM.COM
publisherName=dc=AS270DD,dc=DUEODRF,dc=DE,dc=IBM,dc=COM
```

The first line of each entry is called the distinguished name (DN). DNs are like the complete file name of each entry. Some of the entries do not contain data and are only structural. Those with the line objectclass=inetOrgPerson corresponds to the entries you created for people.

3.6.1 Optionally test the connection to the EIM Domain Controller

Use the **ldapsearch** command to test the connection to the EIM domain controller. This will also provide a sanity check for your EIM configuration.

1. On the command-line interface, enter the command QSH to open a Qshell session.
2. Enter the following commands as shown in Example 3-2 to retrieve a list of all the LDAP entries.

Example 3-2 Result of the ldapsearch command

```
ldapsearch -h as270dd.duedorf.de.ibm.com -p 389 -D cn=administrator  
-w secret -b  
"ibm-eimdomainname=EIM_FFTS,dc=AS270DD,dc=DUEODRF,dc=DE,dc=ibm,dc=com"  
"ibm-eimRegistryName=WebSphereRegistry"
```

(Type this command as one line.)

Where **-h** is the name of the host machine running the LDAP server and **-p** is the according port used by the LDAP server.

-D cn is the LDAP distinguished name or the LDAP administrator and **-w** the password for the LDAP administrator.

-b "ibm-eimdomainname" is the LDAP distinguished name of the EIM domain name entry with the EIM domain parent name, such as **"dc=DUEODRF,dc=DE,dc=ibm,dc=com"** in our sample.

The expected output should look similar to that as shown in Example 3-3.

Example 3-3 Result of LDAP search for the EIM Domain Controller

```
ibm-eimRegistryName=WebSphereRegistry,cn=Registries,ibm-eimdomainnam  
e=EIM_FFTS,dc=AS270DD,dc=DUEODRF,dc=DE,dc=IBM,dc=COM  
objectclass=top  
objectclass=ibm-eimRegistry  
objectclass=ibm-eimSystemRegistry  
objectclass=ibm-eimPolicyListAux  
ibm-eimRegistryName=WebSphereRegistry  
ibm-eimRegistryType=1.3.18.0.2.33.14-caseIgnore  
description=User Registry for WebSphere
```

EIM definitions for SSO with WebSphere

This section describes the EIM components needed to implement SSO with WebSphere and i5/OS. The following components have to be created:

- ▶ Create an EIM registry definition for WebSphere (user registry)
- ▶ Create an EIM registry definition for i5/OS (if not already done). We did this task in 2.1, “Use the EIM Configuration wizard” on page 31.
- ▶ Create an EIM Identifier.
- ▶ For each EIM Identifier
 - Create a source association, type WebSphere (*NOT* Kerberos).
 - Create a target association, type OS/400.

4.1 Create an EIM registry definition for WebSphere

The Identity Token connection factory requires a source User registry definition entry in EIM, which represents the registry that WebSphere is using for authentication. This can be either a local OS registry or an LDAP registry. In our case we create an LDAP registry for WebSphere.

1. Start the iSeries Navigator.
2. Expand **Network** → **Enterprise Identity Mapping** → **Domain Management**.
 - a. Select the EIM domain in which you want to work.
 - b. Type in the password for the administrator and click **OK**. See Figure 4-1.

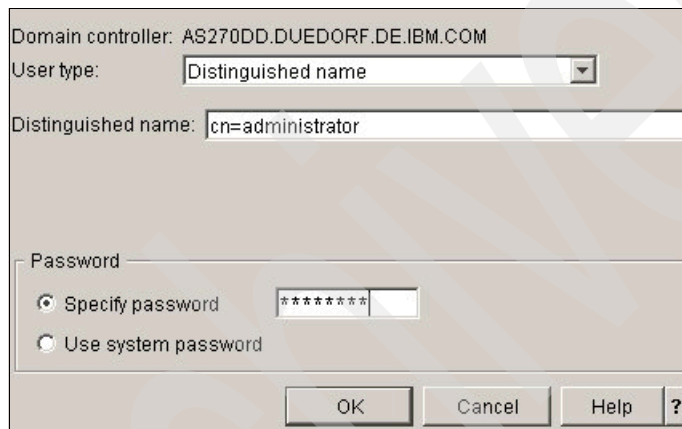


Figure 4-1 EIM Domain Management - login

- c. Expand your EIM domain to which you are now connected.
 - d. Right-click **Add Registry**, and select **System**.

Figure 4-2 shows the Add Registry page of the EIM Domain Management wizard.

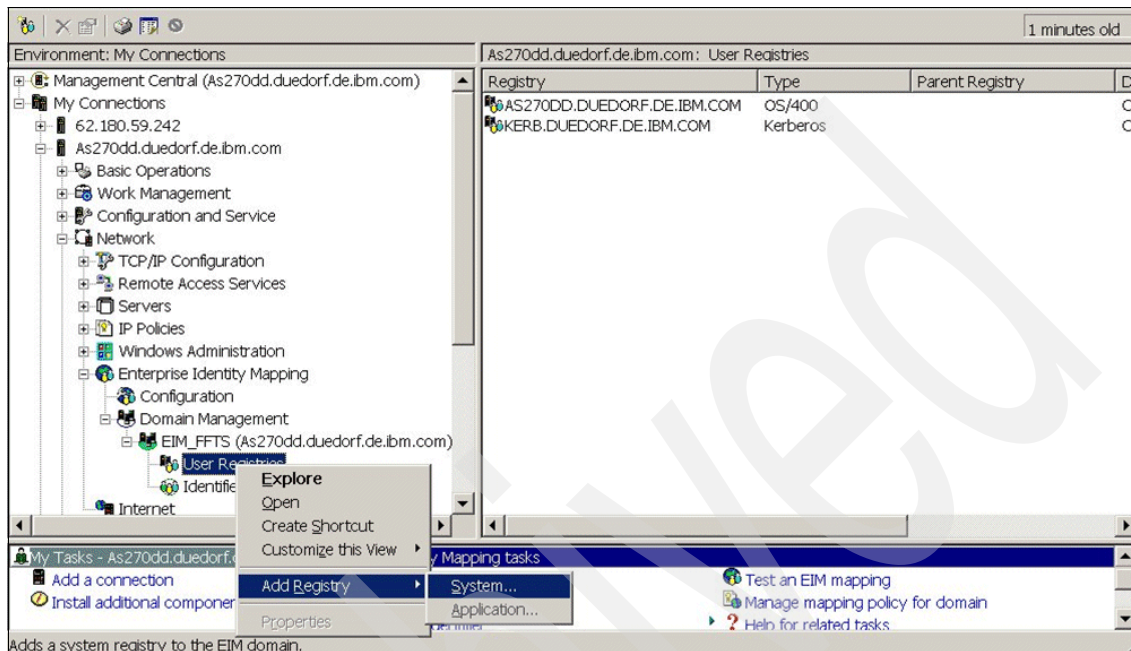


Figure 4-2 EIM Domain Management - Add Registry

In the Registry field, type a name for the registry, in our sample WebSphereRegistry.

- e. If your application server is hosted on a System i5 and configured to use the LocalOS registry for authentication, select **OS/400** as the EIM registry type.

If your application server is configured to use the LDAP registry, select **1.3.18.02.33.14-caselnore** as the EIM registry type.

Note: The value 1.3.18.02.33.14-caselnore is the ObjectIdentifier-normalization form of the registry type whose principals are identified by the LDAP short name attribute. The wizard does not yet handle the descriptive name for this registry type. Support for a descriptive name will be provided in follow-on iSeries Navigator releases.

- f. Type a description in the according field and click **OK**.

Figure 4-3 shows the Create WebSphere Registry page of the EIM Domain Management wizard.

Domain: EIM_FFTS

Registry: WebSphereRegistry

Type: 1.3.18.0.2.33.14-caselnore

Description: User Registry for WebSphere

User registry URL:

Address aliases

Alias:

Type: DNS host name

Add

Alias	Type
-------	------

Remove

OK Cancel Help ?

Figure 4-3 EIM Domain Management - Create WebSphere Registry

4.2 Create an EIM identifier

The Identity Token connection factory requires a user Identifier entry (this is equivalent to an EIM identifier) that represents the user of the application. Create an EIM identifier for each user you want to map to an System i5 user.

To create an EIM identifier, complete the following steps (if you are already connected to the EIM domain, start with step 6):

1. Start iSeries Navigator.
2. Expand **Network** → **Enterprise Identity Mapping** → **Domain Management**.
3. Select the EIM domain in which you want to work.
4. Type in the password for the administrator and click **OK**.
5. Expand your EIM domain to which you are now connected.
6. Right-click **Identifiers** and select **New identifier**.

7. In the New EIM Identifier dialog, provide information about the EIM identifier as follows:
- A name for the identifier; in our sample we define John Day for one of our users.
 - Whether to have the system generate a unique name, if necessary (none in our sample).
 - A description of the identifier.
 - One or more aliases for the identifier, if necessary (none in our sample).
 - After you enter the required information, click **OK** to create the EIM identifier.

Figure 4-4 shows the Create EIM Identifier page of the EIM Domain Management wizard.

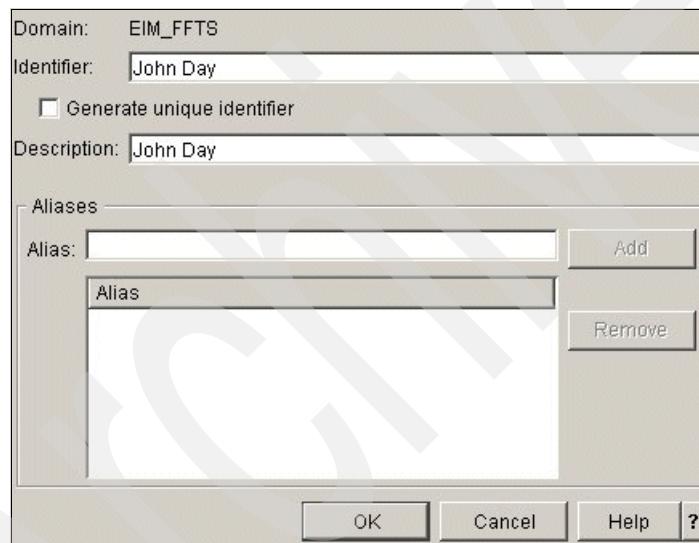


Figure 4-4 EIM Domain Management - Create EIM Identifier

4.3 Create associations

To support mapping from one user ID to another, you need associations for each of your EIM identifiers.

- ▶ A source association to represent the user who authenticate (log in) to WebSphere.
- ▶ A target association to represent the user profile on the target System i5.

Identifier associations allow you to create one-to-one mappings between an EIM identifier and each of the various user identities that are related to the user that the EIM identifier represents.

Note: You can create a policy association to directly define a relationship between multiple user identities in one or more registries and an individual target user identity in another registry. Policy associations use EIM mapping policy support to create many-to-one mappings between user identities without involving an EIM identifier. Policy associations allow you to quickly create a large number of mappings between related user identities in different user registries.

First create the identifier association for the WebSphere registry by completing these steps (if you are already connected to the EIM domain you want, skip steps 1 to 4):

1. Start the iSeries Navigator.
2. Expand **Network** → **Enterprise Identity Mapping** → **Domain Management**.
3. Select the EIM domain in which you want to work.
4. Expand the EIM domain to which you are now connected.
5. Click **Identifiers** to display the list of EIM identifiers for the domain.

Tip: Occasionally, when you attempt to expand the Identifiers folder, it may take a long time before the list of identifiers is displayed. To improve performance when you have a large number of EIM identifiers in the domain, you can “Customize the Enterprise Identity Mapping identifiers view”; for further information, see the iSeries Information Center at:

<http://publib.boulder.ibm.com/iseries/>

6. Right-click the EIM identifier for which you want to create an association and select **Properties** (Figure 4-5 on page 77).

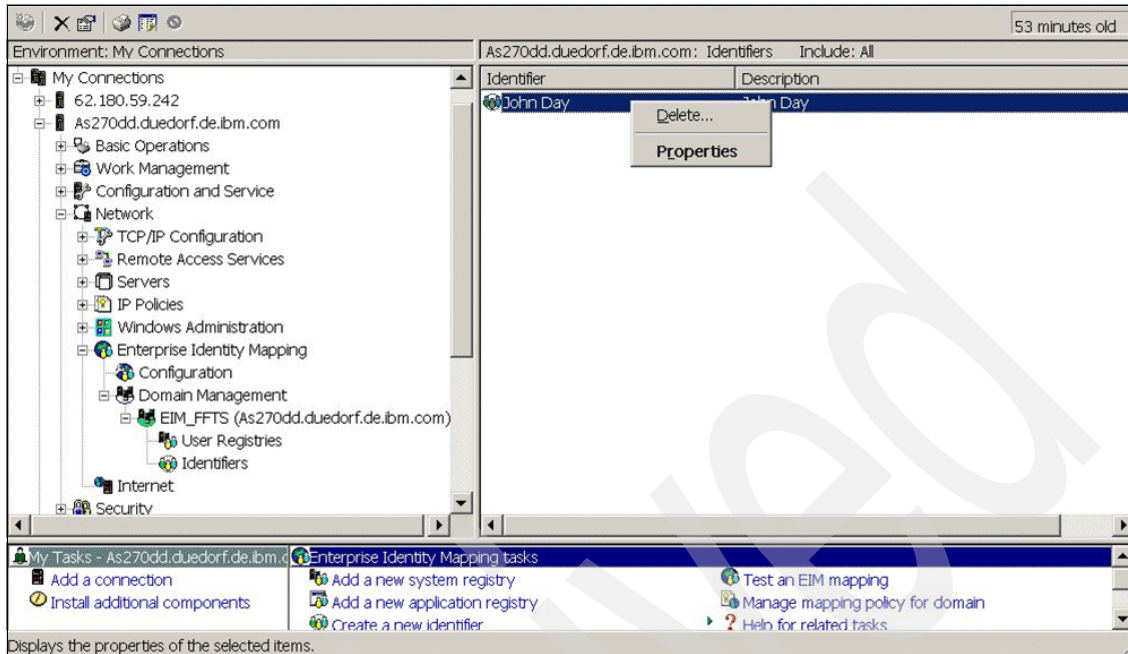


Figure 4-5 EIM Domain Management - Create Association 1

7. In the Add Association dialog, provide information to define the target association, as follows:
 - a. Choose the System i5 registry (the LDAP registry name).
 - b. Type the System i5 user profile, in our sample **JOHND**.
 - c. Select type **Target**.
 - d. After you have provided the required information, click **OK** to create the association.

Figure 4-6 shows the Create Target Association page of the EIM Domain Management wizard.

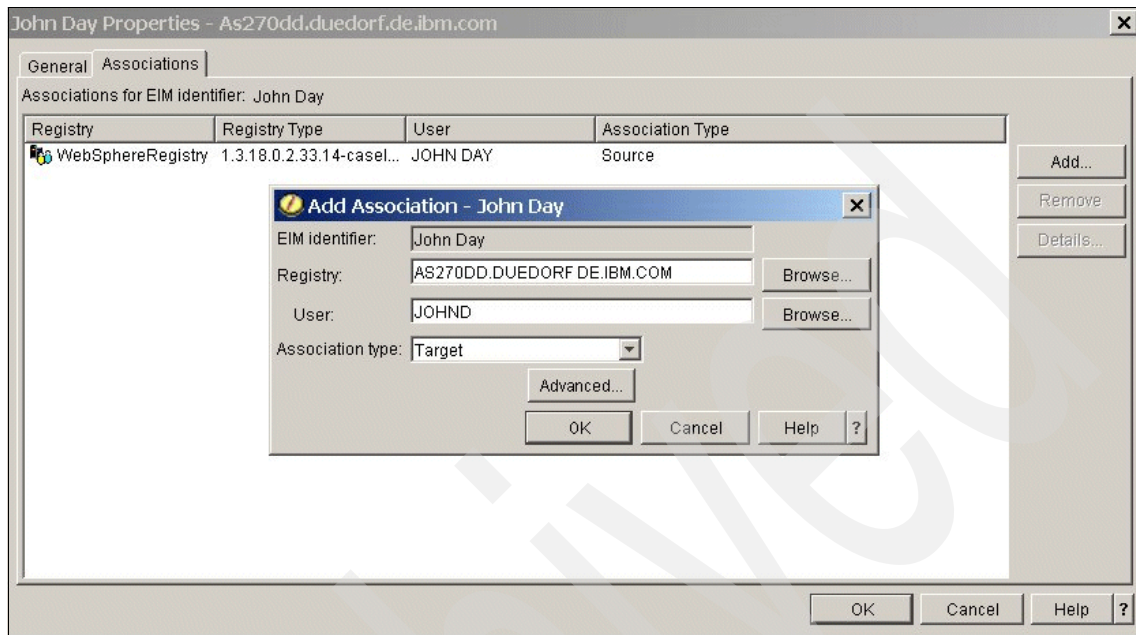


Figure 4-6 EIM Domain Management - Create Target Association

8. Back in the Add Association dialog, provide information to define the source association, as follows:
 - a. The name of the registry that contains the user identity that you want to associate with the EIM identifier. Choose **WebSphereRegistry**.
 - b. The name of the user identity that you want to associate with the EIM identifier, such as John Day.

Tip: WebSphere has changed behavior in principal names being stored in WebSphere Subject objects. Previously, the principal name would have been a short name like 'John Day'.

Now WebSphere prefixes the security realm to the principal name (Format is 'LDAPserver:port/name'), so it looks like this:

'as270dd.duedorf.de.ibm.com:389/John Day'.

You need to configure the EIM association accordingly.

When you have an environment with several different WebSphere Application Server Versions, we recommend creating source associations in both notations (one with the short name of the user and one with the principal name) in our sample as270dd.duedorf.de.ibm.com:389/John Day.

- c. For the type of association, select **Source**.
- d. Click **OK** to create the association. See Figure 4-7.

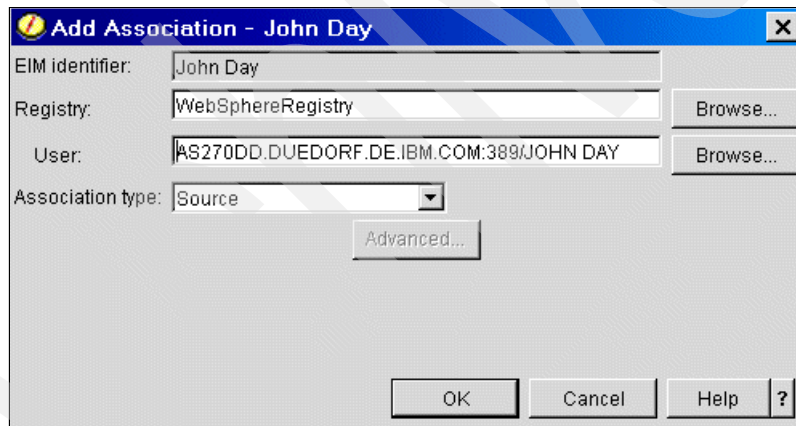


Figure 4-7 EIM Domain Management - Create Source Association

9. As you can see in Figure 4-8, you now have three EIM associations for user John Day.

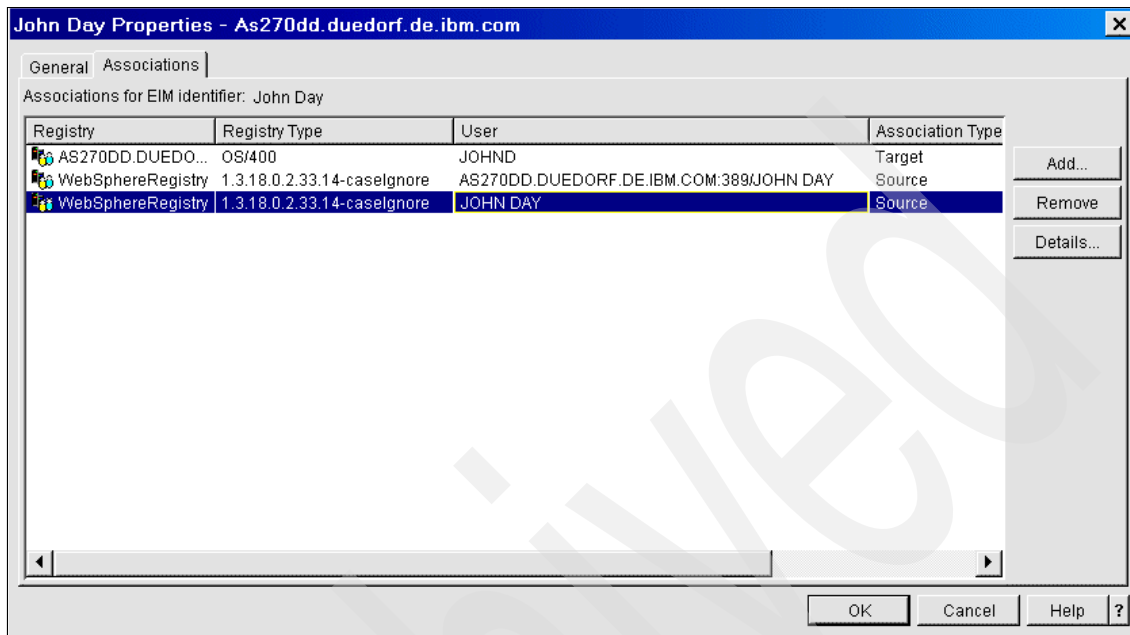


Figure 4-8 EIM Domain Management - EIM associations

4.4 Test EIM mappings

EIM mapping test support allows you to issue EIM mapping lookup operations against your EIM configuration. You can use the test to verify that a specific source user identity maps correctly to the appropriate target user identity. Such testing ensures that EIM mapping lookup operations can return the correct target user identity based on the specified information.

To use mapping test support to test your EIM configuration, complete these steps:

1. Start the iSeries Navigator.
2. Expand **Network** → **Enterprise Identity Mapping** → **Domain Management**.
3. Select the EIM domain in which you want to work.
4. Right-click the EIM domain to which you are connected and select **Test a Mapping....**

5. In the Test a Mapping dialog, specify the following information:
 - a. In Source registry field, provide the registry definition name that refers to the user registry that you want to use as the source of the test mapping lookup operation.
 - b. In Source user field, provide the user identity name that you want to use as the source of the test mapping lookup operation.
 - c. In the Target registry field, provide the registry definition name that refers to the user registry that you want to use as the target of the test mapping lookup operation.
 - d. Optional: In the Lookup information field, provide any lookup information defined for the target user.
 - e. Click **Test** and review the results of the mapping lookup operation when they are displayed.

Figure 4-9 shows the Test a Mapping page of the EIM Domain Management wizard.

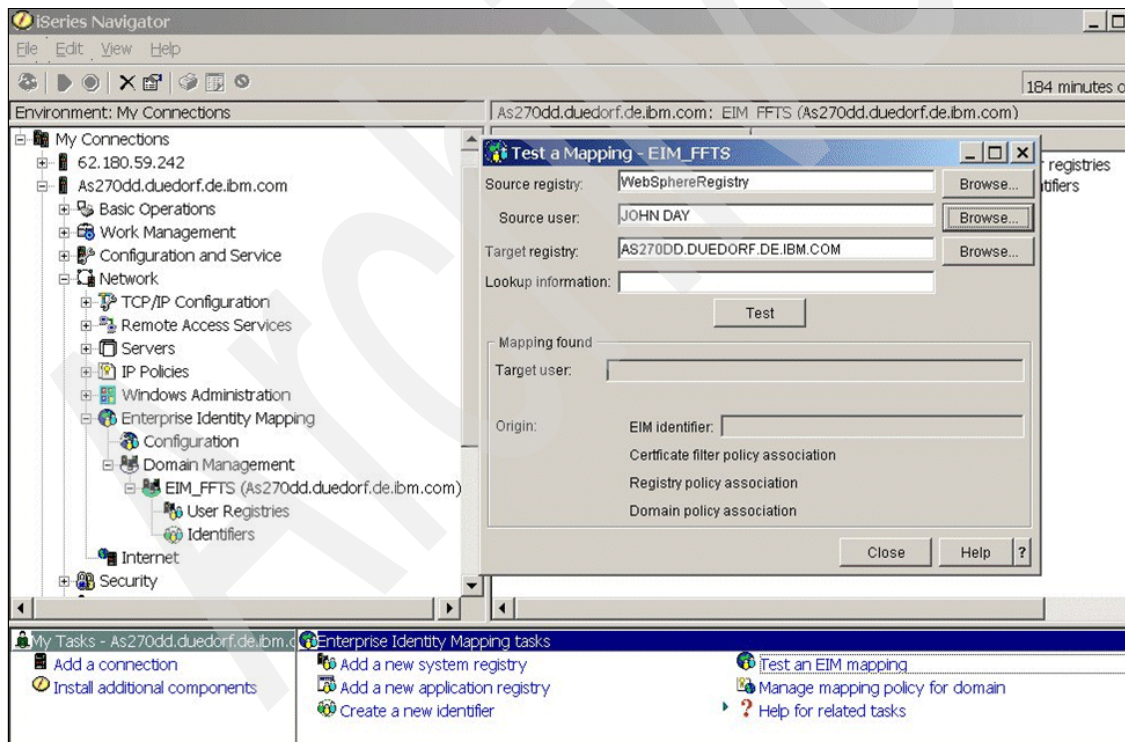


Figure 4-9 EIM Domain Management - Test a Mapping 1

6. If the mapping could be done you will see similar information to that shown in Figure 4-10. Continue testing your configuration, or click **Close** to exit.

Figure 4-10 shows the Test Mapping results page of the WIM Domain Management Wizard.

The screenshot shows a Windows-style dialog box titled "Test a Mapping - EIM_FFTS". It contains several input fields and buttons. The "Source registry" field is set to "WebSphereRegistry", "Source user" is "JOHN DAY", and "Target registry" is "AS270DD.DUEDORF.DE.IBM.COM". Each of these fields has a "Browse..." button to its right. Below these is a "Lookup information:" label followed by an empty text box. A "Test" button is centered below the lookup information. Under the heading "Mapping found", the "Target user:" field displays "JOHND". The "Origin:" section is expanded, showing a checked "EIM identifier:" field with the value "John Day". Below this, three options are listed: "Certificate filter policy association", "Registry policy association", and "Domain policy association". At the bottom right of the dialog are "Close", "Help", and a question mark icon.

Figure 4-10 EIM Domain Management - Test a Mapping results

Create a new WebSphere Application Server profile provided for SSO

In this chapter, we describe how you can create a new WebSphere Application Server instance (profile) where the SSO enabled Web applications will be deployed. You use the IBM Web Administration for iSeries interface that provides a wizard for the creation of an application server, and if you want, a new HTTP server instance.

Of course you can use an already existing WebSphere Application Server profile and implement SSO there. The advantage in creating a new profile through the wizard is that some components needed for SSO implementation inside the application server also will be created.

Parts of these components are the Identity Token Resource Adapter and the associated Connection Factories and some others. This are shown starting in 5.2, "Components needed for SSO" on page 95.

5.1 Create a new WebSphere Application Server

The IBM Web Administration for iSeries interface provides a wizard for creating an application server and, if you want, a new HTTP server instance. You can use this wizard to create a new application server profile that will also have some of the components included that you need to implement SSO.

1. Before you can access the IBM Web Administration for iSeries interface, start the *ADMIN instance of the IBM HTTP Server on System i5. You can do that with the following command:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

You can also use the iSeries Navigator by selecting **Network** → **Servers** and then choosing **TCP/IP**. Highlight **HTTP Administration** in the right frame, right-click it, and select **Start**.

2. To access the IBM Web Administration for iSeries, open a browser and type `http://your.server.name:2001` in the address line (your.server.name is the TCP/IP host.domain name of your System i5). When prompted, sign in using an iSeries user profile that has *ALLOBJ, *SECADM, and *IOSYDCFG special authority.
3. In the iSeries Task window, click the link **IBM Web Administration for iSeries**.
4. In the left link list of the window, click **Create Application Server** link.
5. On the Welcome page, click **Next**.

6. The next page will show you a list of WebSphere Application Server versions installed on your System i5 (Figure 5-1). Select the version of your new profile; in our scenario we use Version 6.0. Click **Next**.

Create Application Server

Select WebSphere Application Server Version

Your system has more than one version of WebSphere Application Server installed.

Choose the type of application server to create

☒ V6.0 WebSphere Application Server V6.0, allows you to add a wide range of complex business logic and dynamic functions to your Web application with a full function J2EETM compliant application server that includes support for Web services and Java Message Service (JMS).

☐ V6.0 ND

☐ V5.1 Base

☐ V5.0 Base

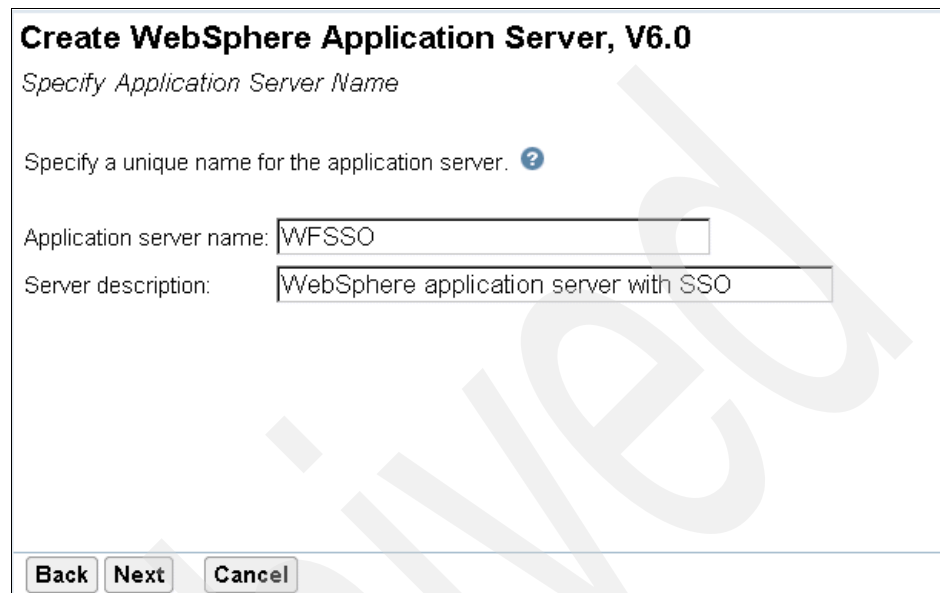
☐ V5.1 Express

☐ V5.0 Express

Back **Next** **Cancel**

Figure 5-1 Create Application Server Wizard- Select Version

7. Give your application server instance a name (Figure 5-2). In our sample, we use WFSSO and type a meaningful description in the Server description input field. Click **Next**.



Create WebSphere Application Server, V6.0

Specify Application Server Name

Specify a unique name for the application server. ?

Application server name:

Server description:

Back **Next** **Cancel**

Figure 5-2 Create Application Server Wizard - Define name

8. In the next window, shown in Figure 5-3, select **Create a new HTTP Server** and click **Next**.

Create WebSphere Application Server, V6.0

Select HTTP Server Type

The application server may be associated with an external HTTP server. If selected, the wizard will set up the external HTTP server with the necessary information to route incoming URL requests to this application server.

Choose the HTTP server type: ?

- ☒ Create a new HTTP server (powered by Apache)
- ☐ Select an existing HTTP server (powered by Apache)
- ☐ Select an existing Domino HTTP server
- ☐ Do not associate an external HTTP server with this application server

Back **Next** **Cancel**

Figure 5-3 Create Application Server Wizard - Define HTTP Server type

9. Type an HTTP server name (Figure 5-4) and a description in the relevant input fields. You can also assign a specific IP address; in our case, we use **All IP addresses**.

We also decided to use 40400 as the port to configure our HTTP server to listen for requests on (use a port number that is not already used in your environment). Click **Next**.

Create WebSphere Application Server, V6.0

Create a new HTTP server (powered by Apache)

A new HTTP server (powered by Apache) will be created and configured to be used by this application server. ?

HTTP server name:

HTTP server description:

Your HTTP server may listen for requests on a specific IP address or on all IP addresses of the system.

On which IP address and TCP port would you like your HTTP server to listen?

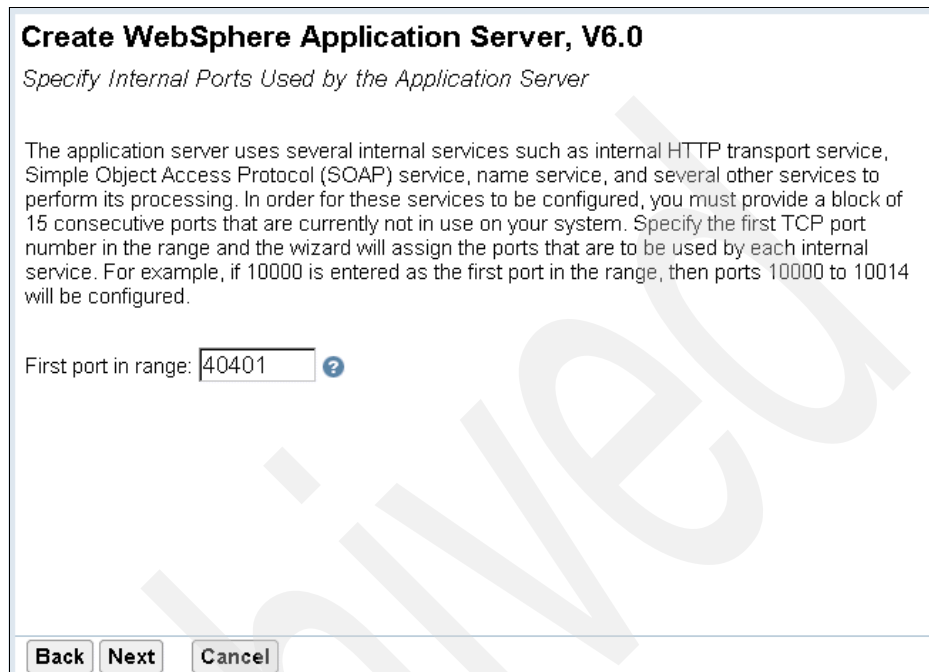
IP address:

Port:

Note: Most browsers make requests to port 80 by default.

Figure 5-4 Create Application Server Wizard - Define HTTP Server Information

10. Type in the first port in the port range for the application server to use; we selected 40401. See Figure 5-5. Click **Next**.



Create WebSphere Application Server, V6.0

Specify Internal Ports Used by the Application Server

The application server uses several internal services such as internal HTTP transport service, Simple Object Access Protocol (SOAP) service, name service, and several other services to perform its processing. In order for these services to be configured, you must provide a block of 15 consecutive ports that are currently not in use on your system. Specify the first TCP port number in the range and the wizard will assign the ports that are to be used by each internal service. For example, if 10000 is entered as the first port in the range, then ports 10000 to 10014 will be configured.

First port in range: ?

Back **Next** **Cancel**

Figure 5-5 Create Application Server Wizard - Specify Internal Ports used by Application Server

11. Decide which sample applications should be installed in the new application server by placing a check in the box next to the option. (Figure 5-6) Click **Next**.

Create WebSphere Application Server, V6.0
Select Business and Sample Applications

You may optionally install business and sample applications into this application server. Choose the applications you want install and the wizard will deploy them for you.

Select which business applications to install:
Note: Installing an IBM business application automatically installs the IBM Welcome Page application.

- ☐ IBM Telephone Directory - An online phone book, providing powerful search capabilities with an easy to use interface.
- ☐ IBM Survey Creator - An online survey tool used to create, configure, and manage Web-based surveys.

Select which sample applications to install:

- ☒ Query - Provides dynamic query service for EJB client applications. This service is accessible only to applications with the appropriate security permissions.
- ☒ Default Applications - A set of samples, including SnoopServlet, that may be used to verify your application server is properly configured.
- ☐ DB2 Web Services - A set of samples that demonstrate the use of Web services Object Runtime Framework to develop Web services applications.

Navigation buttons: Back, Next, Cancel

Figure 5-6 Create Application Server Wizard - Select Sample Applications

12. Before continuing with definition for Identity Tokens, an EIM Domain hosted on an LDAP server has to be configured and running.
Choose **Configure Identity Tokens** and click **Next**.

In the window shown in Figure 5-7, type in the host.domainname of your LDAP server that hosted EIM (in our case AS270DD.DUEDORF.DE.IBM.COM) and the LDAP port (389 is the default port) and also the LDAP administrator DN and password. Click **Next**.

Configure Identity Token SSO for Web to i5/OS Access

Identity Token SSO is a mechanism where a single user signon action permits access to multiple i5/OS servers. This allows your Web-based interfaces to access i5/OS back-end applications without having to prompt for additional authentication. Identity Tokens are implemented using Enterprise Identity Mapping (EIM). EIM maintains the relationships between Web users and i5/OS user profiles. The application server creates a token for the servers configured to support Identity Tokens in this EIM Domain.

Note: EIM is hosted on an LDAP server that must be configured and running before continuing.

Configure Identity Tokens: ?

☐ Do not configure Identity Tokens

☒ Configure Identity Tokens

In order to configure Identity Tokens, an EIM Domain must be configured and the EIM Domain Controller must be running. Specify the LDAP server that is hosting EIM:

LDAP server host name: e.g. "hostname.domain.com"

LDAP port:

LDAP administrator DN: e.g. cn=administrator

LDAP administrator password:

Note: For more information on configuring EIM and Identity Tokens SSO and the required PTFs, see the [iSeries Information Center](#).

Figure 5-7 Create Application Server Wizard - Define LDAP parameters

Select the EIM Domain Name that has been configured as an EIM Domain Controller and the Source Registry Name, which is the name you defined for the EIM registry definition for WebSphere in 4.1, “Create an EIM registry definition for WebSphere” on page 72. Click **Next**. See Figure 5-8.

Create WebSphere Application Server, V6.0
Configure Identity Token EIM Domain Information

The specified LDAP server has been configured as an EIM Domain Controller. Select the correct EIM domain and source registry name that contains the repository of identity information. Identity Tokens will be controlled based on the mappings that have been defined in this location.

Note: In order for the Identity Tokens support to be activated this WebSphere Application Server instance must have global security enabled. Security will need to be manually configured. See the [iSeries Information Center](#) for details.

EIM Domain Name:

Source Registry Name:

Figure 5-8 Create Application Server Wizard - Configure Identity Token EIM Domain Information

Review the Summary pages, notice the three tabs for Application Server, HTTP Server and Identity Tokens. You can view each one of them separately and print the summary using the **Printable Summary** button at the bottom, right of the window. When done, click **Finish**.

Click the **Identity Tokens** tab, shown in Figure 5-9, to see the Summary of Identity Tokens.

Create WebSphere Application Server, V6.0

Summary

When you click **Finish** this WebSphere application server will be created.

Application Server HTTP Server Identity Tokens

EIM Information - LDAP server

LDAP server host name: AS270DD.DJEDORF.DE.IEM.COM
LDAP port: 389

EIM Information - domain and registry

EIM Domain Name: eim fts
Source Registry Name: WebSphereRegistry
Parent DN: dc=as270dd,dc=cuedorf,dc=de,dc=ibm,dc=com

Figure 5-9 Summary of Identity Tokens

You can watch the progress as the server is created in the IBM Web Administration for iSeries window. See Figure 5-10.

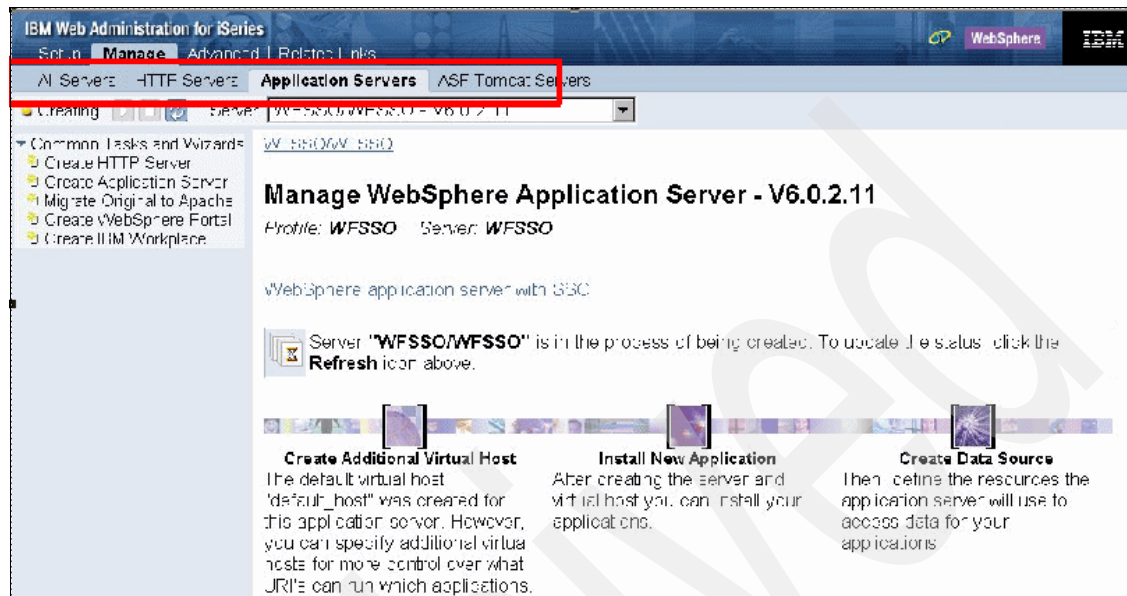


Figure 5-10 Status window

13. Once the application server is created, you can start it by selecting the server name (ours is WFSSO/WFSSO - V6.0.2.11) in the drop-down menu and then clicking the green start icon. See Figure 5-11.



Figure 5-11 Start WebSphere Application Server

5.2 Components needed for SSO

This section shows you which relevant components for SSO are created by using the wizard in 5.1, “Create a new WebSphere Application Server” on page 84.

Because you selected **Configure Identity Tokens** in the wizard and also provided the needed values in the subsequent windows of the wizard, the following components have been configured:

- ▶ J2C Authentication Data Entries
- ▶ Identity Token Resource Adapter
- ▶ J2C Connection Factories

If you want to prepare an already existing WebSphere application profile to implement SSO, these components have to be defined through the WebSphere administrative console using separate steps. You can use the descriptions in this section to create these components.

LDAP and LTPA components are not created by the wizard. You have to define these components through the WebSphere administrator console (see Chapter 6, “Enabling your WebSphere Application Server to use single sign-on” on page 117).

5.2.1 Start the WebSphere administrator console

You can use the WebSphere administrative console to view the components and definitions the wizard created.

There are two ways to start the WebSphere administrator console:

- ▶ From the IBM Web Administration for iSeries (where we left off in the last section), expand **Tools** and click the **Launch Administrative Console** link on the left site of the panel. See Figure 5-12.

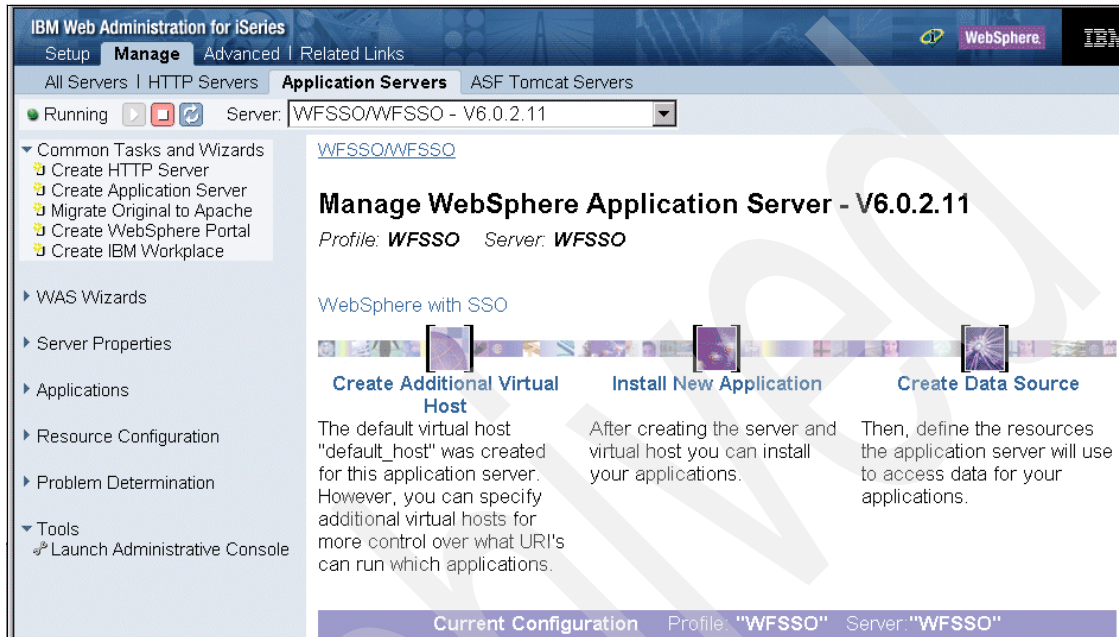


Figure 5-12 Administrative Console launch

- ▶ From your browser

Type the URL for your new WebSphere Application Server profile in the form of `http://servername.domainname:port`; in our sample it is:

`http://as270dd.duedorf.de.ibm.com:40402/ibm/console/`

where *servername.domainname* is the host.domainname of your System i5 and *port* is the administration port the WebSphere Application Server uses.

The port number will be the value you entered for the port range plus 1 in the wizard (we entered 40401 + 1 = 40402; see step 10 on page 89).

Because security is not activated yet, you can log in to the WebSphere administrative console with any user name (no real user on any server).

5.2.2 J2C Authentication Data Entries

Through the Alias name of a J2C Authentication Data Entry, you define a user ID and password for use by the Java 2 Connector Security. Along with this Alias a user and password is stored, which will be used to access resources through a resource adapter, for example, a database. For SSO implementation, the associated idTokenAlias is important. The Alias defines authentication information for the identity token connection factory used for SSO.

In the WebSphere administrator console, expand **Security** → **Global security**. Under Authentication on the right side of the panel, expand **JAAS Configuration** and click **J2C Authentication Data**. Figure 5-13 shows the J2C Authentication for the identity token connection factory page.

Global security

Global security

[Global security](#) > J2EE Connector Architecture (J2C) authentication data entries

Specifies a list of user IDs and passwords for Java 2 connector security to use.

▣ Preferences

New Delete

Select	Alias ↕	User ID ↕	Description ↕
<input type="checkbox"/>	idTokenAlias	cn=administrator	authentication information for identity token connection factory

Total 1

Figure 5-13 J2C Authentication for identity token connection factory

Click the link **idTokenAlias** to see the configured values for the Alias name. The User ID represents the administrator of the LDAP server, including the password. See Figure 5-14.

The screenshot shows a web browser window with the title "Global security". The breadcrumb navigation is "Global security > J2EE Connector Architecture (J2C) authentication data entries > idTokenAlias". Below the breadcrumb, a description states: "Specifies a list of user IDs and passwords for Java 2 connector security to use." The "Configuration" tab is selected. Under the "General Properties" section, there are four fields: "Alias" with the value "idTokenAlias", "User ID" with the value "cn=administrator", "Password" with masked characters "••••••", and "Description" with the value "authentication information for i". At the bottom of the configuration area are four buttons: "Apply", "OK", "Reset", and "Cancel".

Figure 5-14 2C Identity token connection factory-Authentication Data Entries

5.2.3 Identity Token Resource Adapter

In general, a resource adapter represents a Resource Adaptor Archive (RAR) file containing code that implements a library for connecting with an Enterprise Information System(EIS) back end, such as iSeries, CICS®, SAP®, or People Soft.

RAR is a file that contains all the information necessary for installing, configuring, and running a JCA Resource Adapter. A resource adapter is a system-level software driver that is used by a Java application to connect to an EIS. A resource adapter plugs into an application server and provides connectivity between the EIS, the application server, and the enterprise application.

We need the resource adapter to create the identity token; see 1.6, “Identity Token Resource Adapter” on page 13.

1. In the WebSphere administrator console, expand **Resource** → **Resource Adapters**. You will see the **Identity Token Connector** resource, which was created by the wizard (Figure 5-15). The resource adapter contains configuration properties that are defined in the J2C specification.



Figure 5-15 Resource Adapter created by the wizard

Also important is the scope where the resource adapter is created. A resource can be visible in the administrative console collection table at the cell, node, cluster, or server level. By changing the value for Scope, you can see other variables that apply to a resource and might change the contents of the collection table. As you see, this Identity Token Adapter has been created on the node level (see Figure 5-15 on page 99).

Resource adapters

A resource adapter is an implementation of the J2EE Connector Architecture Specification that provides access for applications to resources outside of the server or provides access for an Enterprise Information System (EIS) to applications on the server. It can provide application access to resources such as DB2, CICS, SAP and PeopleSoft. It can provide an EIS with the ability to communicate with message driven beans that are configured on the server. Some resource adapters are provided by IBM; however, third party vendors can provide their own resource adapters. A resource adapter implementation is provided in a resource adapter archive file; this file has an extension of .rar. A resource adapter can be provided as a standalone adapter or as part of an application, in which case it is referred to as an embedded adapter. Use this pane to install a standalone resource adapter archive file. Embedded adapters are installed as part of the application install.

☒ Scope: Cell=AS270DD_wfsso1, Node=AS270DD_wfsso1

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#)

Cell

→ Node

Server

Figure 5-16 Resource Adapter created by the wizard- Scope Node

2. Click the **Identity Token Connector** link. You will see the Resource Adapter Archive file (RAR) parameters for the idTokenRA.rar or idTokenRA.JCA15.rar file that has been installed by the wizard. See Figure 5-17.

Resource adapters

A resource adapter is an implementation of the J2EE Connector Architecture Specification that provides access for applications to resources outside of the server or provides access for an Enterprise Information System (EIS) to applications on the server. It can provide application access to resources such as DB2, CICS, SAP and PeopleSoft. It can provide an EIS with the ability to communicate with message driven beans that are configured on the server. Some resource adapters are provided by IBM; however, third party vendors can provide their own resource adapters. A resource adapter implementation is provided in a resource adapter archive file; this file has an extension of .rar. A resource adapter can be provided as a standalone adapter or as part of an application, in which case it is referred to as an embedded adapter. Use this panel to install a standalone resource adapter archive file. Embedded adapters are installed as part of the application install.

☐ Scope: Cell=**AS270DD_wfsso**, Node=**AS270DD_wfsso**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#)

Cell

→ Node

Server

Figure 5-17 Resource Adapter Identity Token Connector

There are two EIM Identity Token Connection Factory RAR files available: one that implements the J2EE Connector Architecture (JCA) Version 1.0 and one that implements JCA Version 1.5.

- The RAR file for JCA V1.0 has the name idTokenRA.rar.
- The RAR file for JCA V1.5 has the name idTokenRA.JCA15.rar.

Both RAR files are located in the following System i IFS directories:

- OS/400 V5R2: /QIBM/ProdData/OS400/Java400/ext/.
- OS/400 V5R3 and later: /QIBM/ProdData/OS400/security/eim/.

Note: Note that the JCA V1.5 RAR file should be used for WebSphere Application Server Version 6.0 or later. The JCA V1.0 RAR file can be used for WebSphere Application Server V5.0, V5.1, and V6.0.

To obtain JCA V1.5 version of the EIM Identity Token Connection Factory, apply the appropriate PTF:

- For V5R2: 5722SS1 SI24296
- For V5R3: 5722SS1 SI20620
- For V5R4: 5722SS1 SI23261

Important: At the time this document was written, the RAR file for JCA V1.0 was installed by the wizard. You see it in the name of the Archive path and Class path used when you look at the Identity Token Connector window. If you see that in your environment, reinstall the resource adapter to use the idTokenRA.JCA15.rar resource adapter, as described in 5.2.5, “Reinstall resource adapter” on page 108.

5.2.4 Connection factories

The following connection factories have been created by the wizard. We look at the definition in the following way:

1. In the WebSphere administrator console, expand **Resource** → **Resource Adapters**. You will see the Identity Token Connector resource that was created by the wizard. Click the link **Identity Token Connector**.

2. Under Additional Properties, click the link **J2C connection factories** in the right site of the panel. Two connection factories, CF1 and CF2 have been created by the wizard. See Figure 5-18.

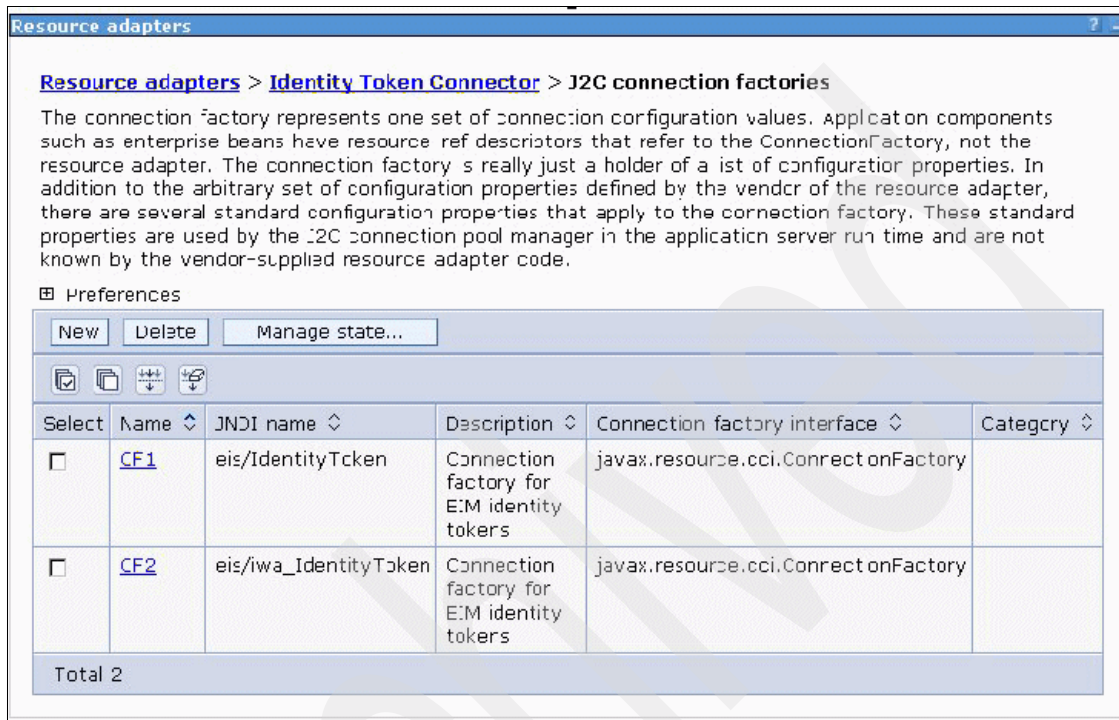


Figure 5-18 J2C Connection Factories

3. We now have a look at the definitions of the CF1 Connection Factory. CF2 is identical to CF1 except the JNDI name for CF2 is different (eis/iwa_IdentityToken). Click the link **CF1**. Figure 5-19 shows the first window of the Connection Factory CF1's parameters. Click **OK** to proceed to the next window.

Resource adapters

[Resource adapters](#) > [Identity Token Connector](#) > [J2C connection factories](#) > **CF1**

The connection factory represents one set of connection configuration values. Application components such as enterprise beans have resource-ref descriptors that refer to the ConnectionFactory, not the resource adapter. The connection factory is really just a holder of a list of configuration properties. In addition to the arbitrary set of configuration properties defined by the vendor of the resource adapter, there are several standard configuration properties that apply to the connection factory. These standard properties are used by the J2C connection pool manager in the application server run time and are not known by the vendor-supplied resource adapter code.

Configuration

General Properties

- * Scope: cells:AS270DD_wfss01:nodes:AS270DD_wfss01
- * Name: CF1
- JNDI name: eis/IdentityToken
- Description: Connection factory for EIM identity tokens

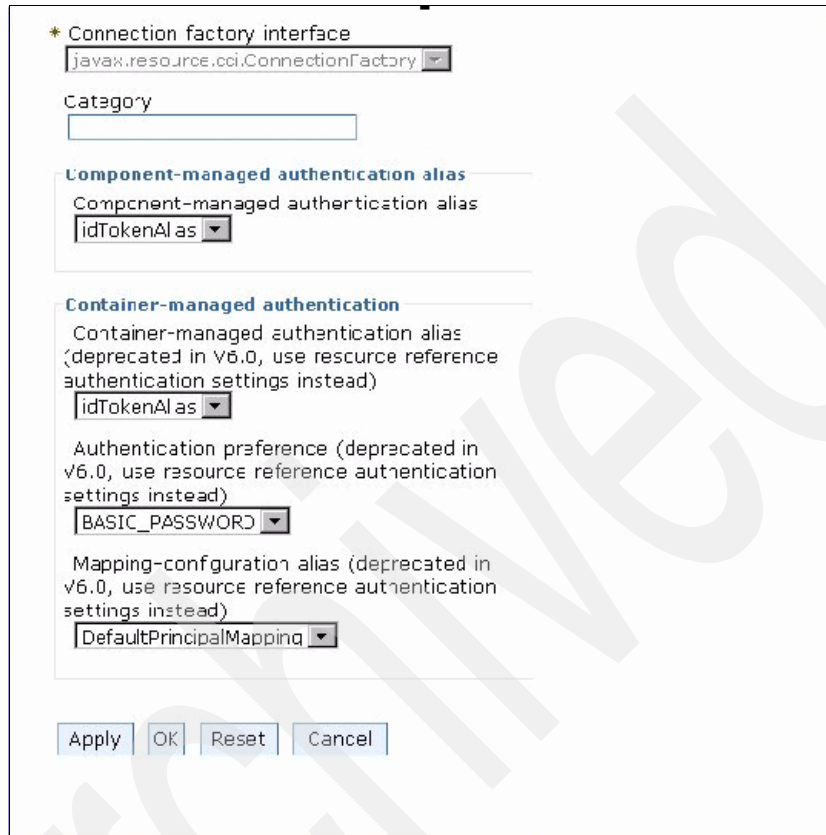
Additional Properties

- [Connection pool properties](#)
- [Advanced connection factory properties](#)
- [Custom properties](#)

Related Items

Figure 5-19 Connection Factory CF1 parameters - Part 1

Figure 5-20 shows the second page of Connection Factory CF1 parameters. Click **OK** to exit.



* Connection factory interface
javax.resource.cci.ConnectionFactory

Category

Component-managed authentication alias
idTokenAlias

Container-managed authentication
Container-managed authentication alias
(deprecated in V6.0, use resource reference authentication settings instead)
idTokenAlias

Authentication preference (deprecated in v6.0, use resource reference authentication settings instead)
BASIC_PASSWORD

Mapping-configuration alias (deprecated in v6.0, use resource reference authentication settings instead)
DefaultPrincipalMapping

Apply OK Reset Cancel

Figure 5-20 Connection Factory CF1 parameters - Part 2

- Now look at the additional properties of the CF1 Connection Factory. Click **Custom Properties** under the Additional Properties area. The Custom Properties window (Figure 5-21 on page 107) shows the properties needed by the resource provider and resource factories. The more important properties on this page are LdapHostName (the host name of the LDAP server, in our case, AS270DD.DUEDORF.DE.IBM.COM), LdapHostPort (in our case the default port 389), EimDomainName (eim_ffts in our case), parent domain (dc=as270dd,dc=duedorf,dc=de,dc=ibm,dc=com), and SourceRegistryName (WebSphere Registry).

The correct value for EimDomainName is eim_ffts and for the parent domain of the EIM controller is dc=as270dd,dc=duedorf,dc=de,dc=ibm,dc=com. These need to be adjusted to fit your environment. If you have to reinstall the resource adapter, you can follow the steps described in 5.2.5, “Reinstall resource adapter” on page 108. That section describes how the properties for the Connection Factory are set correctly.

Resource adapters > **Identity Token Connector** > **J2G connection factories** > **CF1** > **Custom properties**

Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.

☒ Preferences

Name	Value	Description	Required
LdapHostName	AS270DD.DUEDORF.DE.IBM.COM	Required. The fully qualified TCP/IP host name of the LDAP server hosting the EIM domain controller. For example: myLDAPServer.com	false
LdapHostPort	389	Optional. The port number of the LDAP server.	false
EimDomainName	eim_ffts.dc=as270dd.dc=duedorf.dc=de.dc=ibm.dc=com	Required. The simple (undistinguished) name of the EIM domain this resource adapter is using. For example: Auth Tokens Domain	false
ParentDomain		Optional. The LDAP DN value for the parent domain of the EIM domain this resource adapter is using. For example: dc=myServer, dc=myCompany, dc=com	false
SourceRegistryName	WebSphereRegistry	Required. The name of the EIM registry in which the authenticated username has a source mapping.	false
KeyTimeoutSeconds	1200	Optional. The number of seconds that the identity token is valid for.	false
KeySize	512	Optional. The number of bits to be used for the identity token's key.	false
UseSSL		Optional. Indicates whether SSL is to be used when connecting to the LDAP server. Default is false.	false

Figure 5-21 Custom properties

5.2.5 Reinstall resource adapter

Before reinstalling the adapter, record the settings currently used for the adapter and the adapter's connection factories. Write down the values of the properties so you can reuse them when installing the new adapter with the idTokenRA.JCA15.rar. The following values are needed:

- ▶ The name of your Identity Token Resource Adapter (Identity Token Connector)
- ▶ For each Identity Token Connection Factory:
 - The name of the connection factory.
 - Record the values for each General Property.
- ▶ Under Additional Properties, click **Custom properties** and record the value for each Custom Property.

Note: To find these values, follow the steps described in 5.2.3, “Identity Token Resource Adapter” on page 98 and 5.2.4, “Connection factories” on page 102. Then go back to this point to reinstall the adapter.

To reinstall the adapter:

1. Delete the resource adapter (Identity Token Connector). This action cascades down and results in the deletion of all the adapter's connection factories.
 - a. Navigate to the Resource Adapters window by selecting **Resources** → **Resource Adapters**.
 - b. Select the Node where your Identity Token Resource Adapter is installed, and click **Apply**.
 - c. Select the name of your Identity Token Resource Adapter and click **Identity Token Connector**.
 - d. Click **Delete**.
 - e. Save your changes to the master configuration.
2. Configure a new Identity Token adapter.
 - a. Navigate to the Resource Adapters panel by selecting **Resources** → **Resource Adapters**.
 - b. Select the **Node** (scope) where your Identity Token Resource Adapter should be installed. You can use the **Browse Nodes** button to select the node. After the selection, click **Apply**.
 - c. Click the **Install RAR** button. See Figure 5-22 on page 109.

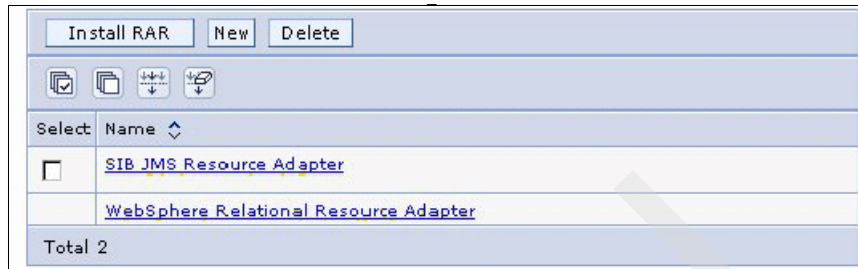


Figure 5-22 Install RAR

- d. In the next window, use the **Browse** button under the Local path to navigate to the System i5 IFS and select the **idTokenRA.JCA15.rar** file. For OS/400 V5R3 and later, you find it in the /QIBM/ProdData/OS400/security/eim directory. For OS/400 V5R2, you should find it in /QIBM/ProdData/OS400/Java400/ext/.

Be sure that under Scope, the node name for the WebSphere Application Server profile is selected. Click **Next**. See Figure 5-23.

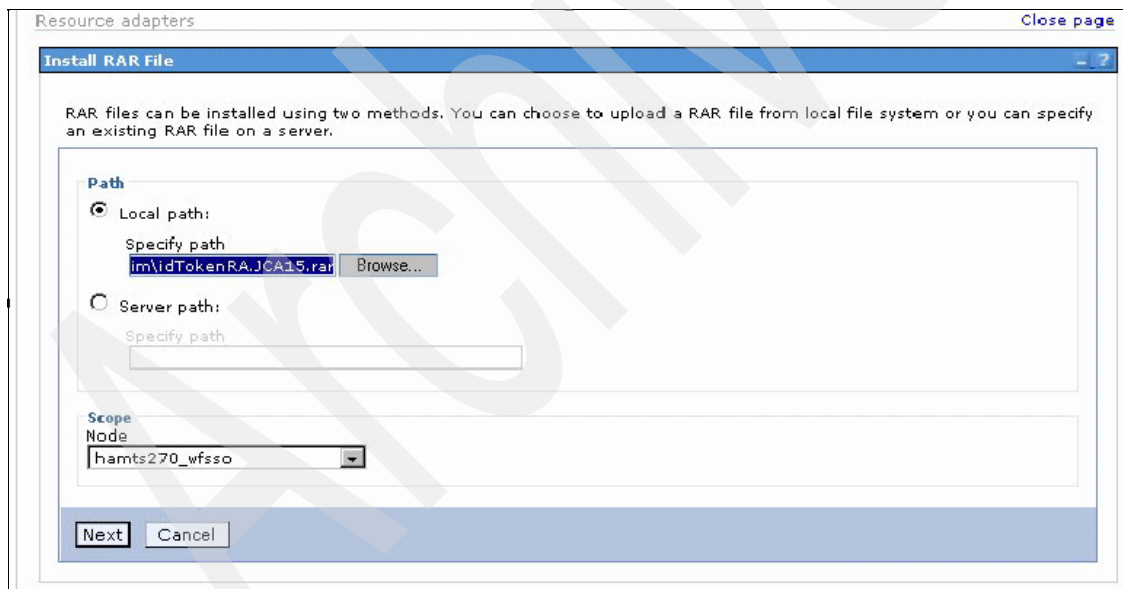


Figure 5-23 Install RAR Path

- e. In the next window, specify the property settings you recorded earlier for the resource adapter. For the Name, use Identity Token Connector. Leave all other properties blank. Click **OK**.

The Archive Path and Classpath is set automatically.

3. Now you create the two connection factories inside the Identity Token Connector, with the name CF1 and CF2 (the same names that were created by the wizard before).
 - a. Navigate to the Resource Adapters window by selecting **Resources** → **Resource Adapters**. Click the **Identity Token Connector** link.
 - b. Under Additional Properties, click the link **J2 connection factories** on the right side of the window.
 - c. Click the **New** button.
 - d. Fill in the properties for:

Name: CF1

JNDI name: eis/IdentityToken

Description: A meaningful name

Component-managed authentication alias: idTokenAlias

Container-managed authentication alias: idTokenAlias

The idTokenAlias has been created in 5.2.2, “J2C Authentication Data Entries” on page 97.

Leave the default values for the rest of the properties. Click **Apply**.

Figure 5-24 shows the CF1 Connection Factory Properties page.

General Properties	Additional Properties
<p>* Scope</p> <p>cells:AS270DD_wfssso:nodes:AS270DD_wfssso</p>	<ul style="list-style-type: none"> Connection pool properties Advanced connection factory properties Custom properties
<p>* Name</p> <p>CF1</p>	
<p>JNDI name</p> <p>eis/IdentityToken</p>	
<p>Description</p> <p>CF for SSO sample</p>	
<p>* Connection factory interface</p> <p>javax.resource.cci.ConnectionFactory</p>	
<p>Category</p> <p></p>	
<p>Component-managed authentication alias</p> <p>Component-managed authentication alias</p> <p>idTokenAlias</p>	
<p>Container-managed authentication</p> <p>Container-managed authentication alias (deprecated in V6.0, use resource reference authentication settings instead)</p> <p>idTokenAlias</p> <p>Authentication preference (deprecated in V6.0, use resource reference authentication settings instead)</p> <p>None</p>	

Figure 5-24 CF1 Connection Factory Properties

- e. Click the link **Custom properties** under Additional Properties. You will see a list with the additional properties, as shown in Figure 5-25.

[Resource adapters](#) > [Identity Token Connector](#) > [J2C connection factories](#) > [CF1](#) > **Custom properties**

Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.

☐ Preferences

Name	Value	Description	Required
eimDomainName			false
keySize	512		false
keyStoreName			false
keyStorePassword			false
keyTimeoutSeconds	1200		false
ldapHostName			false
ldapHostPort	389		false
parentDomain			false
sourceRegistryName			false
trustStoreName			false
trustStorePassword			false
useSSL	false		false
Total 12			

Figure 5-25 CF1 Connection Factory - Custom Properties

- f. Now change the values for the properties by clicking the appropriate link, for example, **eimDomainName**.

- g. In the next window, type the value (in our case, `eim_ffts`) in the value field and click **OK**. See Figure 5-26.

Resource adapters > Identity Token Connector > J2C connection factories > CF1 > Custom properties > eimDomainName

Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.

Configuration

General Properties

* Scope
cells:AS270DD_wfsso:ncdes:AS270DC_wfsso:servers:wfsso

☐ Required

Name
eimDomainName

Value
eim_ffts

Description

Type
java.lang.String

Apply OK Reset Cancel

Figure 5-26 CF1 Connection Factory Custom Properties - eimDomainName

- h. Do the same steps for updating the values for:
- ldapHostName
as270dd.duedorf.de.ibm.com
 - parentDomain (the parent Domain of the EIM Domain)
dc=as270dd,dc=duedorf,dc=de,dc=ibm,dc=com
 - sourceRegistryName - WebSphere Registry
- You have created this sourceRegistry as described in 4.1, “Create an EIM registry definition for WebSphere” on page 72.

At the end of your updates, the Custom properties should look similar to those shown in Figure 5-27.

Resource adapters > Identity Token Connector > J2C connection factories > CF1 > Custom properties

Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.

⊞ Preferences

Name	Value	Description	Required
eimDomainName	eim ffts		false
keySize	512		false
keyStoreName			false
keyStorePassword			false
keyTimeoutSeconds	1200		false
ldapHostName	as270dd.duedorf.de.ibm.com		false
ldapHostPort	389		false
parentDomain	dc=as270dd,dc=duedorf,dc=de,dc=ibm,dc=com		false
sourceRegistryName	WebSphere Registry		false
trustStoreName			false
trustStorePassword			false
useSSL	false		false
Total 12			

Figure 5-27 CF1 Connection Factory - Custom Properties Final

- i. Repeat the same steps to create the CF2 Connection Factory beginning with step a on page 110.
- j. Save your changes to the master configuration and restart your application server so that the changes will become effective.

5.2.6 Trace capabilities of the Identity Token Connection Factory

The JCA V1.5 version of the Identity Token Connection Factory includes trace capabilities that can be enabled in the following way:

1. From the WebSphere Administrative Console, select **Servers** → **Application Servers** → **server_name** → **Change Log Details Levels**.
2. Select the **Runtime** tab.
3. Select the **Save runtime changes to configuration** and check the box as well.

4. Remove any previous entries in the text field and type the following:
`com.ibm.jca.idtoken.*=all:com.ibm.eim.token.*=all`
5. Apply and save the changes.

Archived

Enabling your WebSphere Application Server to use single sign-on

In this chapter we cover the steps required to enable your WebSphere Application Server profile (instance) for single sign-on (SSO) security:

- ▶ Define the LDAP settings.
- ▶ Define LTPA settings.
- ▶ Enable Global Security for your WebSphere Application Server.
- ▶ Optionally, create another J2C Connection Factory for the WebFacing application.

You will be using the Global Security wizard of the WebSphere administrative console to configure your WebSphere Application Server.

6.1 Defining the LDAP settings for your WebSphere Application Server

In this section, you will update the settings in the LDAP User Registry and the Advanced LDAP Settings pages to match those in your LDAP registry. The attributes we set previously in our LDAP directory for our WebSphere administrator named WASADMIN1 are shown in the Figure 6-1.

Name	Value	Type
cn	WASADMIN1	text attribute
departmentnumber	2005	text attribute
mail	wasadmin1@de.ibm.com	text attribute
objectclass	top	text attribute
objectclass	inetOrgPerson	text attribute
objectclass	organizationalPerson	text attribute
objectclass	person	text attribute
sn	WASADMIN1	text attribute
telephonenumber	123456	text attribute
userpassword	7B 53 48 41 7D DA F3 59 F2 1B 2D 26 5D F4 D9 46	password [SHA]

Figure 6-1 WASADMIN1 LDAP attributes

We assume that you still have the WebSphere administrator console open; otherwise, start this interface as described in 5.2.1, “Start the WebSphere administrator console” on page 95.

1. In the navigation bar on the left pane, expand **Security** and click **Global Security** (Figure 6-2).



Figure 6-2 WebSphere Application Server - Console navigation

2. Under User registries, click **LDAP**. On the LDAP User Registry page, set the following properties:

- a. Server User ID

Provides the user ID of the administrator for the WebSphere administrative domain. In our case we use the value WASADMIN1, which is the user we created before. See 3.5, “Create a user for the WebSphere Administrator” on page 65.

When Global Security is active and you start the WebSphere Application Server administrative console, you are prompted to log in with an administrative account. This account is checked against the LDAP directory, and the new user must be found in the LDAP directory.

In the Server User ID field, we have to enter the Distinguished Name (DN) for our WebSphere administrator user. In our case, we specify `cn=WASADMIN1,cn=employees,dc=AS270DD,dc=DUEDORF,dc=DE,dc=IBM,dc=COM`.

- b. Server User Password

The password corresponds to the Server User ID field. This field is case sensitive.

- c. Type

Do not change this value. See the explanations in step 4 on page 121.

- d. Host

The fully qualified DNS name of the machine on which the LDAP directory runs. You should use the full domain name, in our case, `AS270DD.DUEDORF.DE.IBM.COM`.

- e. Base Distinguished Name (DN)

Specifies the distinguished name for the application server to use when binding to the directory service. If no name is specified, the application server binds anonymously. This field is not case sensitive.

In our sample, we define:

`DC=AS270DD,DC=DUEDORF,DC=DE,DC=IBM,DC=COM`.

- f. Bind Distinguished Name

The DN of the user who is capable of performing searches on the directory. In most cases, this field is not required; typically, all users are authorized to search an LDAP directory. However, if the LDAP directory contents are restricted to certain users, you need to specify the DN of an authorized user, for example, an administrator, `cn=administrator`.

We specify here the DN of our WebSphere administration user, cn=WASADMIN1,cn=employees,dc=AS270DD,dc=DUEDORF,dc=DE,dc=IBM,dc=COM.

g. Bind Password

The password corresponding to the Bind Distinguished Name field. This value is required only if you specified a value for the Bind Distinguished Name field. This field is case sensitive.

h. Ignore Case

By default, WebSphere Application Server does a case sensitive comparison for authorization. This implies that a user who is authenticated by Domino should match exactly the entry (including the base distinguished name) in the WebSphere Application Server authorization table. If case sensitivity should not be considered for the authorization, the Ignore Case property should be enabled in the LDAP user registry settings.

Check **Ignore Case**.

Figure 6-3 on page 121 shows the LDAP User Registry page.

Global security > LDAP User Registry

Uses the LDAP user registry settings when users and groups reside in an external LDAP directory. When security is enabled and any of these properties are changed, go to the Global Security panel, located under Security in the left navigation menu. Click Apply or OK to validate the changes.

Configuration

General Properties	Additional Properties
<p>* Server user ID <input type="text" value="cn=WASADMIN1,cn=employee"/></p> <p>* Server user password <input type="password" value="*****"/></p> <p>Type <input type="text" value="Custom"/></p> <p>* Host <input type="text" value="AS270DD.DUEDORF.DE.IBM.C"/></p> <p>Port <input type="text" value="389"/></p> <p>Base distinguished name (DN) <input type="text" value="DC=AS270DD,DC=DUEDORF,"/></p> <p>Bind distinguished name (DN) <input type="text" value="cn=WASADMIN1,cn=employee"/></p> <p>Bind password <input type="password" value="*****"/></p> <p>Search timeout <input type="text" value="1202"/> seconds</p> <p><input checked="" type="checkbox"/> Reuse connection</p> <p><input checked="" type="checkbox"/> Ignore case for authorization</p> <p><input type="checkbox"/> SSL enabled</p> <p>SSL configuration <input type="text" value="AS270DD_wfsso/DefaultSSLSettings"/></p>	<ul style="list-style-type: none"> ■ Advanced Lightweight Directory Access Protocol (LDAP) user registry settings ■ Custom properties

Figure 6-3 LDAP User Registry page

3. Click **Apply** to apply the updates.
4. Next we will configure the Advanced LDAP settings. Click **Advanced Lightweight Directory Access Protocol (LDAP) user registry settings**. The Advanced LDAP Setting page is used, when users and groups reside in an external LDAP directory. Default values for all the users and the group related filters are already completed in the appropriate fields.

You can change these values depending on your requirements. These default values are based on the type of LDAP server selected in the LDAP User Registry page. If this type changes (for example from Netscape to Secureway), the default filters automatically change. When the default filter values change, the LDAP server type changes to Custom to indicate that custom filters are used.

a. User Filter

Specifies an LDAP filter clause for searching the registry for users.

Set the User Filter according the values in your LDAP configuration. In our case we define here:

```
(&(cn=%v)(objectclass=inetOrgPerson))
```

b. Group Filter

Specifies the LDAP group filter that searches the user registry for groups. This option is typically used for Security Role to Group assignments. It specifies the property by which to look up groups in the directory service. For more information about this syntax, see the LDAP directory service documentation.

Set the Group Filter to the according value, in our case:

```
(&(cn=%v)(|(objectclass=groupOfNames)(objectclass=groupOfUniqueNames)(objectclass=groupOfURLs)))
```

c. User ID Map

Specifies the LDAP filter that maps the short name of a user to an LDAP entry.

Set the User ID Map to the according value, in our case:

```
*:cn
```

d. Group ID Map

Specifies the piece of information that represents groups when groups appear. For example, to display groups by their names, specify *:cn. The asterisk (*) is a wildcard character that searches on any object class in this case. This field takes multiple objectclass:property pairs delimited by a semicolon (;).

Set the Group ID Map to the according value, in our case:

```
*:cn
```


e. Group Member ID Map

Specifies the LDAP filter which identifies user to group relationships.

Set the Group Member ID Map to the according value, in our case:

`(&(cn=%v)(objectclass=groupOfNames))`

f. Let the values for the rest of the properties be their default. Click **OK**.

Figure 6-4 shows the Advanced LDAP settings page.

Global security ? -

Global security > LDAP User Registry > Advanced Lightweight Directory Access Protocol (LDAP) user registry settings

Specify advanced LDAP User Registry settings when users and groups reside in an external LDAP directory. When security is enabled and any of these advanced settings are changed, go to the Global Security panel, located under Security in the left navigation panel. Click Apply or OK to validate the changes.

Configuration

General Properties

User filter

Group Filter

User ID map

Group ID map

Group member ID map

☐ Perform a nested group search

Certificate map mode

Certificate filter

Figure 6-4 Advanced LDAP Settings

5. Save the changes.

6.2 Define the LTPA properties

From the WebSphere administrative console, navigate to the Global security page by selecting **Security** → **Global security**.

1. Under Authentication, expand **Authentication mechanisms** and click **LTPA**.
2. Under Additional Properties, click **Single signon (SSO)**. See Figure 6-5. Single sign-on is enabled by default. If it has been disabled, click **Enabled**.

Global security > LTPA

Specifies the Lightweight Third Party Authentication (LTPA) configuration settings. When security is enabled and any of these properties are changed, go to the Global Security panel, located under Security in the left navigation menu. Click Apply or OK to validate the changes. The LTPA keys are automatically generated the first time that you enable security. After you enable security, WebSphere Application Server can generate a new set of keys in two ways. If you need to change the password, make the change and click OK or Apply to generate the keys. You do not need to click Generate Keys. If you do not need to change the password, click Generate Keys. The new set of keys are not used until they are saved.

Configuration

Generate Keys Import keys Export Keys

General Properties

* Password

* Confirm password

* Timeout
120

Key file name

Apply OK Reset Cancel

Additional Properties

■ [Single signon \(SSO\)](#)

■ [Trust association](#)

Figure 6-5 LTPA configuration page

3. On the SSO configuration page, shown in Figure 6-6 on page 125, enter the fully qualified name of the DNS domain for which single sign-on is effective. This parameter will define the scope of effectiveness for your configuration. For example, if your domain is defined as `ibm.com`, single sign-on will work between the domains `rochester.ibm.com` and `austin.ibm.com`, but not `austin.otherCompany.com`.

[Global security](#) > [LTPA](#) > **Single signon (SSO)**

Specifies the configuration values for single signon.

Configuration

General Properties

☒ Enabled

☐ Requires SSL

Domain name

☒ Interoperability Mode

☒ Web inbound security attribute propagation

Figure 6-6 LTPA Single signon page

The cookie is sent for all of the servers that are contained within the domains that you specify in this field. If the domain name is not fully qualified, WebSphere Application Server does not set a domain name value for the LtpaToken cookie and the single sign-on is only valid for the server that created the cookie. In our case, we use duedorf.de.ibm.com.

Note: The domain field is optional and if left blank the Web browser defaults to the domain name of the SSO cookie, which is the WebSphere Application Server that created it. This behavior may be desirable when you have defined multiple virtual hosts and each virtual host needs its own or separate domain to be specified in the single sign-on cookie.

You can configure the Domain name field using any of the following values:

- (Blank)
- A single domain name, for example: rochester.ibm.com
- UseDomainFromURL
- Multiple domain names, for example: rochester.ibm.com;austin.ibm.com
- Multiple domain names and UseDomainFromURL

If you specify the UseDomainFromURL value type, WebSphere Application Server sets the single sign-on domain name value to the domain of the host that makes the request. For example, if an HTTP request comes from server1.rochester.ibm.com, WebSphere Application Server sets the single sign-on domain name value to rochester.ibm.com.

The value, UseDomainFromURL, is case insensitive. You can type usedomainfromurl to use this value.

When you specify multiple domains, you can use the following delimiter: a semicolon (;), a space (), a comma (,), or a pipe (|). WebSphere Application Server searches the specified domains in the order from left to right. Each domain is compared with the host name of the HTTP request until the first match is located. For example, if you specify ibm.com;rochester.ibm.com and a match is found in the ibm.com domain first, WebSphere Application Server does not continue to search for a match in the rochester.ibm.com domain. However, if a match is not found in either the ibm.com or rochester.ibm.com domains, then WebSphere Application Server does not set a domain for the LTPA Token cookie.

4. Enable the Interoperability mode option if you want to allow single sign-on connections in WebSphere Application Server to operate with previous versions of the application server.
5. Click **OK** and you will be taken back to the LTPA settings page automatically.

6.2.1 LTPA keys

Before you exit the LTPA settings page, you also need to configure the LTPA keys, which are used by the administrative domain that you are configuring. You must perform one of the following steps, based on the number of administrative domains you are configuring:

- ▶ If you are configuring the first or only WebSphere Application Server administrative domain, generate the LTPA keys using these steps:
 - a. Type the LTPA password to be associated with these LTPA keys in the Password and Confirm Password fields. You must use this password when importing these keys into other WebSphere Application Server administrative domain configurations (if any) and when you configure single sign-on for Domino.
 - b. Click **Generate Keys** to generate keys for LTPA.
 - c. Click **OK** and you will go back to the Global security window automatically.

- If you are configuring an additional WebSphere Application Server administrative domain, you must *import* the LTPA keys used during the configuration of the first administrative domain (see 6.2.3, “Importing LTPA keys” on page 129).

Note: The LTPA keys are automatically generated the first time security is enabled. After security is enabled, a new set of keys can be generated in two ways. If the password needs to be changed, change the password and click **OK** or **Apply** to generate the keys (no need to press the Generate Keys button). If password is the same, just press the **Generate Keys** button. The new set of keys will not be used until saved.

Figure 6-7 shows the LTPA settings page.

[Global security](#) > **LTPA**

Specifies the Lightweight Third Party Authentication (LTPA) configuration settings. When security is enabled and any of these properties are changed, go to the Global Security panel, located under Security in the left navigation menu. Click Apply or OK to validate the changes. The LTPA keys are automatically generated the first time that you enable security. After you enable security, WebSphere Application Server can generate a new set of keys in two ways. If you need to change the password, make the change and click OK or Apply to generate the keys. You do not need to click Generate Keys. If you do not need to change the password, click Generate Keys. The new set of keys are not used until they are saved.

Configuration

General Properties	Additional Properties
<p>* Password</p> <p>* Confirm password</p> <p>* Timeout 120</p> <p>Key file name</p>	<p><input type="checkbox"/> Single signon (SSO)</p> <p><input type="checkbox"/> Trust association</p>

Figure 6-7 LTPA settings page

6.2.2 Exporting LTPA keys

To enable SSO support in the WebSphere Application Server across multiple WebSphere Application Server domains (cells) the LTPA keys and the password need to be shared among each of the domains. The time stamps on the domains should be similar to prevent the tokens from appearing as expired between the cells. You will only perform these steps for one WebSphere Administrative Server even if you are configuring SSO for use with multiple WebSphere Application Server administrative domains. This file is subsequently used during the configuration of additional administrative domains and during the configuration of SSO for Domino.

For security purposes, the exported keys are encrypted with a user-defined password. The same password is needed when importing the keys into another cell.

The Export Keys button from the LTPA settings page can be used to export the LTPA keys to other domains or cells. Complete the following steps in the administrative console to export key files for LTPA.

1. Navigate in the WebSphere navigation menu to **Security** → **Global security**.
2. Under Authentication, expand **Authentication mechanisms** and click **LTPA**. This will open the page shown in Figure 6-8 on page 129.
3. In the Key File Name field, specify the name and location of the file (in the iSeries integrated file system (IFS)) where the keys are to be stored. You can use any file name and extension. For this example we used /QIBM/UserData/LTPA_Key/ltpakey_wpsadmin02.txt.

Global security > LTPA

Specifies the Lightweight Third Party Authentication (LTPA) configuration settings. When security is enabled and any of these properties are changed, go to the Global Security panel, located under Security in the left navigation menu. Click Apply or OK to validate the changes. The LTPA keys are automatically generated the first time that you enable security. After you enable security, WebSphere Application Server can generate a new set of keys in two ways. If you need to change the password, make the change and click OK or Apply to generate the keys. You do not need to click Generate Keys. If you do not need to change the password, click Generate Keys. The new set of keys are not used until they are saved.

Configuration

Generate Keys Import keys Export Keys

General Properties

* Password
 Password field with masked characters (dots).

* Confirm password
 Confirm Password field with masked characters (dots).

* Timeout
 Timeout field with value 120.

Key file name
 Key file name field with value VQIBM/UserData/LTPA_Key/lt

Apply OK Reset Cancel

Additional Properties

Single signon (SSO)
 Trust association

Figure 6-8 LTPA Settings page - Export LTPA keys

4. Click the **Export Keys** button. A file is created with the LTPA keys in it. Exporting keys will fail if a new set of keys was generated or imported and not saved prior to exporting. To avoid this failure, make sure you save the new set of keys before you export them.
5. Click **OK**.

6.2.3 Importing LTPA keys

Importing keys is a dynamic operation. All the servers that are running at this time are updated with the new set of keys and any back-level tokens signed with the back-level keys fail validation and the user is prompted to log in again.

To import LTPA keys, perform these steps in the WebSphere administrative console:

1. Navigate in the WebSphere navigation menu to **Security** → **Global security**.

2. Under Authentication, expand **Authentication mechanisms** and click **LTPA**.
3. Change the password in the password fields to match the password in the cell from which you are importing the keys.
4. Click **Save** to save the new set of keys in the repository. This is an important step to be completed before importing the keys. If the password and the keys do not match, the servers fail to start. In that case, you would have to turn off security and complete this process again.
5. In the Key File Name field, enter the full path of the file where the keys are stored.
6. Click **Import Keys**. The keys are now imported into the system.
7. Click **OK**.
8. Click **Save** to save the new set of keys in the repository. It is important to save the new set of keys to match the new password so that the servers start.

Note the name and extension you specify; you must use this file when you configure a single sign-on for any additional WebSphere Application Server administrative domains and for Domino.

Note: The directories into which you want to export the Key file must have permission for the iSeries user profile your application server from which you want to export is running (QEJBSVR is the default user profile).

9. Click **Save** to save the file.

6.3 Enable Global Security for your WebSphere Application Server

The term *Global Security* represents the security configuration that is effective for the entire *security domain*. The basic requirement for a security domain is that the access ID returned by the registry from one server be the same access ID as that returned from the registry on any other servers within the same security domain.

Global security applies to all applications running in the environment. It determines whether security is used at all, the type of registry against which authentication takes place, the type of authentication mechanism, and some other security values.

An authentication mechanism in WebSphere is responsible for creating a credential, which is an internal product representation of a successfully authenticated client user. The abilities of the credential are determined by the configured authentication mechanism.

1. Navigate in the WebSphere navigation menu to **Security** → **Global security**, if you are not already there.
2. Check **Enable global security** and uncheck **Enforce Java 2 Security**.
3. Verify that the Cache Timeout field is set to a reasonable value for your application. When the timeout value is reached, WebSphere Application Server clears the security cache and rebuilds the security data. If the value is set too low, the extra processing overhead can be unacceptable. If the value is set too high, you create a security risk by caching security data for a long period of time. The default value is 600 seconds.
4. Active Protocol. This is the active authentication protocol for Remote Method Invocation over the Internet Inter-ORB Protocol (RMI IIOP) requests, used when security is enabled. WebSphere Application Server can be configured to support both Common Secure Interoperability versions 2 (CSIV2) and the IBM Secure Authentication Service (IBM SAS). IBM SAS is the authentication protocol used by all releases of WebSphere Application Server prior to Version 5. The CSIV2 has been defined by the Object Management Group (OMG) as a standard authentication protocol so that vendors can interoperate securely. The implemented CSIV2 in WebSphere Application Server has more features than the IBM SAS. If all the servers within the security domain are WebSphere Version 5 or 6 servers, specify the CSI protocol. If some servers are 3x or 4x servers, specify CSI and SAS.

Select **CSI** for the Active Protocol.

5. Active authentication mechanism specifies the active authentication mechanism when security is enabled. The active authentication mechanism is not configurable. Also, this version of the product only supports Lightweight Third Party authentication (LTPA).

6. Specify **LDAP** for the Active User Registry and click **OK**. See Figure 6-9.

Global security

Specifies the global security configuration for a managed domain. The following steps are required to turn on security: 1. Configure the desired user registry listed under User registries and set its properties. 2. Select the Enable global security option on this panel. 3. Select the configured user registry type from the Active user registry option on this panel.

Configuration

General Properties

- ☒ Enable global security
- ☐ Enforce Java 2 security
- ☐ Enforce fine-grained JCA security
- ☐ Use domain-qualified user IDs
- * Cache timeout: 500 seconds
- ☒ Issue permission warning
- Active protocol: CSI
- Active authentication mechanism: LDAP
- Active user registry: Lightweight Directory Access Protocol (LDAP) user registry
- ☐ Use the Federal Information Processing Standard (FIPS)

Apply OK Reset Cancel

User registries

- Custom
- LDAP**
- Local OS

Authentication

- Authentication mechanisms
- Authentication protocol
- JAAS Configuration

Authorization

- Authorization providers

Additional Properties

- Custom properties

Figure 6-9 WebSphere Application Server - Global Security

7. When security is configured, validation against the user registry settings are done. An attempt is made to authenticate the server ID to the configured user registry. Validating the user registry settings after enabling global security can avoid problems when you restart the server for the first time. If any mismatches are found, you will get a message similar to that shown in Figure 6-10 on page 133. Correct the settings in the LDAP configuration windows and try to apply the general security again.

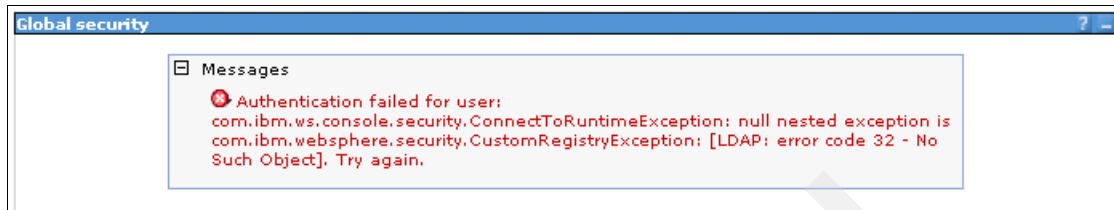


Figure 6-10 Authentication failed message

6.4 Configure a shared library for the jt400.jar file

The EIM Identity Token Connection Factory requires `eim.jar` to be in the classpath for the connection factory, and `jt400.jar` must be in the classpath for the sample application.

Note: In the Network Deployment environment, configure the `eim.jar` and `jt400.jar` files for the WebSphere node where the connection factory is to be configured.

For OS/400 V5R3, JTOpen Version 4.3 of the `jt400.jar` file is already on your iSeries system. However, you still need to configure a shared library for the `jt400.jar` file. Use the WebSphere Administrative Console to create a shared library for the `jt400.jar` file using these steps:

1. From the WebSphere Administrative Console expand **Environment** and click **Shared Libraries**.
2. Be sure that you have set the scope to the node where you wish to create the shared library.
3. Click **New**.

Figure 6-11 shows the Shared Libraries page.



Figure 6-11 Shared Libraries

4. Type the name of the shared library in the Name field; we used jt400SharedLib.
5. Type the full path name of the jt400.jar file in the Classpath field. For OS/400 V5R3, this is /QIBM/ProdData/HTTP/public/jt400/lib/jt400.jar.
6. Click **OK**.

Figure 6-12 shows the shared Libraries for the jt400.jar Classpath.

The screenshot shows the 'Shared Libraries' configuration window. At the top, it says 'Shared Libraries > New'. Below this, a description states: 'Specifies a container-wide shared library that can be used by deployed applications.' The 'Configuration' tab is selected. Under the 'General Properties' section, there are three fields: 'Name' with the value 'jt400SharedLib', 'Description' (empty), and 'Classpath' with the value '/Q1BM/ProdData/HTTP/public/jt400/lib/jt400.jar'. There is also a 'Native Library Path' field which is empty. At the bottom, there are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'.

Figure 6-12 Shared Libraries for jt400.jar Classpath

6.4.1 Create an application class loader

An application class loader for the shared library makes the jt400.jar file available to all applications deployed on the application server.

1. In the WebSphere Administrative Console, expand **Servers** and click **Application Server**.
2. Click the name of the server to which the class loader is to be added.

Figure 6-13 shows the List of Application Servers page.

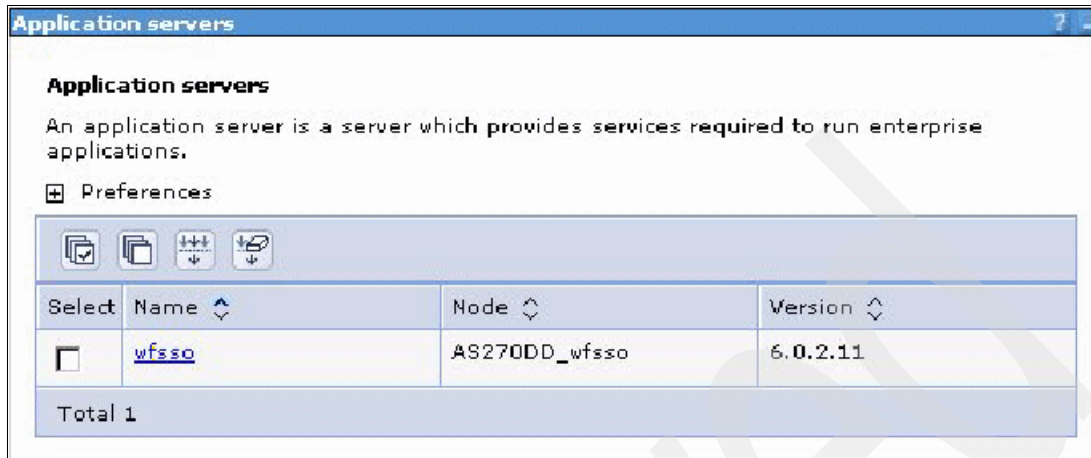


Figure 6-13 List of Application Servers

3. Click the **Configuration** tab.
4. Under Server Infrastructure, expand **Java and Process Management**.
5. Click **Class loader**.
6. Click **New**.
7. Click **OK** (let Classloader Mode default to PARENT_FIRST).

Figure 6-14 on page 137 shows the Application Class Loader page.



Figure 6-14 Application Class Loader

8. Click the Classloader ID for the classloader just created.

Figure 6-15 shows the Application Class Loader Properties page.

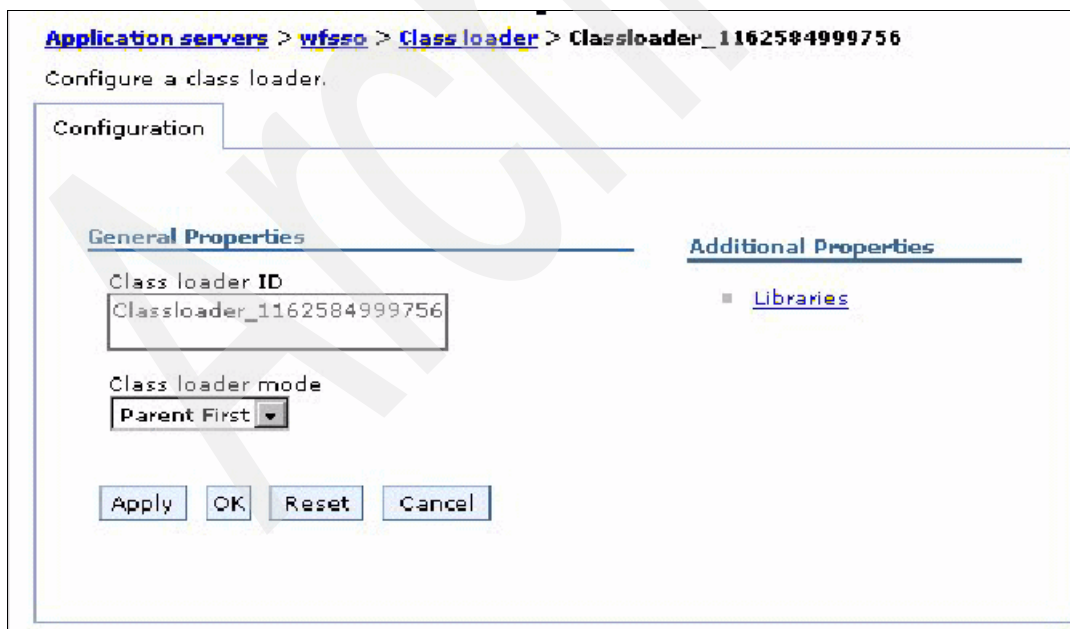


Figure 6-15 Application Class Loader Properties

9. Click **Libraries**.
10. Click **Add**.
11. Select the name of the shared library you created.
12. Click **OK**.

Resource adapters

A resource adapter is an implementation of the J2EE Connector Architecture Specification that provides access for applications to resources outside of the server or provides access for an Enterprise Information System (EIS) to applications on the server. It can provide application access to resources such as DB2, CICS, SAP and PeopleSoft. It can provide an EIS with the ability to communicate with message driven beans that are configured on the server. Some resource adapters are provided by IBM; however, third party vendors can provide their own resource adapters. A resource adapter implementation is provided in a resource adapter archive file; this file has an extension of .rar. A resource adapter can be provided as a standalone adapter or as part of an application, in which case it is referred to as an embedded adapter. Use this pane to install a standalone resource adapter archive file. Embedded adapters are installed as part of the application install.

☒ Scope: Cell=AS270DD_wfsso1, Node=AS270DD_wfsso1

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#)

Cell
AS270DD_wfsso1

→ Node
AS270DD_wfsso1

Server

Figure 6-16 Application Class Loader Library

13. Save your changed settings to the master configuration.

6.4.2 Configuring an additional connection factory

Because we will have several applications deployed to our WebSphere application profile, we decided to create a separate connection factory for every application. Here we create the CF3 connection factory for our Order Entry application.

You can use the WebSphere Administrative Console to manually configure the connection factory, or you can use a Jacl script to automatically configure the connection factory. We use the WebSphere Administrative Console to create an additional J2C Connection Factory:

1. From the WebSphere Administrative Console, expand **Resources** and click **Resource Adapters**.
2. Click the link **Identity Token Connector**. This resource adapter that has been configured through the wizard for you.
3. Under Additional Properties, click the link **J2CConnection Factories**.
4. In the next window, you can already see the created CF1 and CF2 Connection Factories. Click **New**. See Figure 6-17.

[Resource adapters](#) > [Identity Token Connector](#) > J2C connection factories

The connection factory represents one set of connection configuration values. Application components such as enterprise beans have resource-ref descriptors that refer to the ConnectionFactory, not the resource adapter. The connection factory is really just a holder of a list of configuration properties. In addition to the arbitrary set of configuration properties defined by the vendor of the resource adapter, there are several standard configuration properties that apply to the connection factory. These standard properties are used by the J2C connection pool manager in the application server run time and are not known by the vendor-supplied resource adapter code.

⊞ Preferences

☐ ☐ ☐ ☐

Select	Name	JNDI name	Description	Connection factory interface	Category
<input type="checkbox"/>	CF1	eis/IdentityToken	Connection factory for EIM identity tokens	javax.resource.cci.ConnectionFactory	
<input type="checkbox"/>	CF2	eis/iwa_IdentityToken	Connection factory for EIM identity tokens	javax.resource.cci.ConnectionFactory	

Total 2

Figure 6-17 List of Connection Factories for Identity Token Connector

5. Enter a name for the factory. In our sample, we use CF3. Enter eis/idTokenRoot in the JNDI name field. We will later use this name when we define the JNDI name in the WebSphere binding section of the Web Deployment Descriptor; see 7.5, “Define resource reference for both applications” on page 163 for the Order Entry application. See Figure 6-18.

Resource adapters

[Resource adapters](#) > [Identity Token Connector](#) > [J2C connection factories](#) > **CF3**

The connection factory represents one set of connection configuration values. Application components such as enterprise beans have resource-ref descriptors that refer to the ConnectionFactory, not the resource adapter. The connection factory is really just a holder of a list of configuration properties. In addition to the arbitrary set of configuration properties defined by the vendor of the resource adapter, there are several standard configuration properties that apply to the connection factory. These standard properties are used by the J2C connection pool manager in the application server run time and are not known by the vendor-supplied resource adapter code.

Configuration

General Properties	Additional Properties
<p>* Scope</p> <p>cells:AS270DD_wfsso:nodes:AS270DD_wfsso</p>	<p>■ Connection pool properties</p>
<p>* Name</p> <p>CF3</p>	<p>■ Advanced connection factory properties</p>
<p>JNDI name</p> <p>eis/idTokenRoot</p>	<p>■ Custom properties</p>
<p>Description</p> <p>CF for WF</p>	<p>Related Items</p>
<p>* Connection factory interface</p> <p>javax.resource.cci.ConnectionFactory</p>	
<p>Category</p> <p></p>	

Figure 6-18 Creating new Connection Factory CF3

6. In the Container-managed and Component-managed authentication aliases, select **idTokenAlias**. You created this alias in 5.2.2, “J2C Authentication Data Entries” on page 97. Click **OK**. See Figure 6-19.

The dialog box is titled "Component-managed authentication alias". It contains two main sections:

- Component-managed authentication alias:** A dropdown menu showing "idTokenAlias".
- Container-managed authentication:** This section includes:
 - A label: "Container-managed authentication alias (deprecated in V6.0, use resource reference authentication settings instead)".
 - A dropdown menu showing "idTokenAlias".
 - A label: "Authentication preference (deprecated in V6.0, use resource reference authentication settings instead)".
 - A dropdown menu showing "BASIC_PASSWORD".
 - A label: "Mapping-configuration alias (deprecated in V6.0, use resource reference authentication settings instead)".
 - A dropdown menu showing "(none)".

At the bottom of the dialog are four buttons: "Apply", "OK", "Reset", and "Cancel".

Figure 6-19 Creating new Connection Factory CF3 Authentication Alias

7. Now we have three connection factories, as shown in Figure 6-20.



Figure 6-20 List of Connection Factories for Identity Token Connector New

8. Click the link of the just created connection factory **CF3**. Under Additional Properties, click **Custom Properties**.
9. Click the link **LdapHostName**. In the value field, type in the name of the LDAP server. In our sample, it is AS270DD.DUEDORF.DE.IBM.COM. Click **OK**.
10. In the Custom Properties window:
 - a. Click the link **EimDomainName**. In the value field, type in the name of the EIM Domain server. In our sample, it is EIM_FFTS. Click **OK**.
 - b. Select the **ParentDomain** link. In the value field, type in the LDAP DN value for the parent domain. In our example, it is dc=AS270dd,dc=duedorf,dc=de,cd=ibm,cd=com. Click **OK**.
 - c. Click **SourceRegistryName**. In the value field, type **WebSphereRegistry**. Click **OK**.
11. Save and close the WebSphere Administrative Console and restart the WebSphere Application Server.

Prepare your applications to use single sign-on and EIM

This chapter describes and demonstrates how to use the IBM WebSphere Development Studio Client (WDSC) for iSeries Version 6.0 to exploit the iSeries password-elimination SSO strategy.

It explains the tasks required to configure both a Web Tools and WebFacing application to participate in an SSO environment.

Suppose that John Day is a customer-service representative with MyCompany, a toys supplier. All of MyCompany's applications run on the shop's single iSeries server, and two of the core applications, Order Entry and Customer Inquiry, have been recently Web-enabled using both the WebFacing and Web Tools packaged in the WDSC. John is excited about now being able to access these applications from a Web browser; however, he finds the need to log on to each application separately cumbersome and time consuming. To help ease John's pain, the IT department is implementing SSO so he will only have to log on once when using both applications.

We use the Order Entry and the Customer Inquiry (Web Tools-interaction tutorial) applications shipped with WDSC. Both applications are Web-enabled versions of existing RPG applications, though the Customer Inquiry application was created using the iSeries Web Tools and the Order Entry was created using the iSeries WebFacing tool.

For both applications, the RPG business logic and data reside on a single iSeries server; they take user input from a Web page, call an RPG procedure, and return an output Web page with data. The values entered on the input page are passed to an iSeries host program for processing and the values returned from the host program are displayed in the output page.

All information and source code that you need to work with on this application are provided as samples that are available in the WDSC product and has to be installed on a workstation, where you do application development. You get this information when you start your IBM WebSphere Development Studio Client (WDSC) for iSeries and then:

- ▶ For the Order Entry application, select **Help** → **Samples Gallery** → **Application samples** → **Web** → **iSeries Web** → **WebFacing Tools**.
- ▶ For the Customer Inquiry application, select **Help** → **Tutorials Gallery** → **Do and Learn Creating an iSeries Web interaction**.

Enabling WebSphere security does not automatically secure the applications it is running, so securing the application is the next step.

Setting up security for your application includes the following tasks:

- ▶ Setting up security roles and constraints for your application by editing the Web Deployment Descriptor file web.xml.
- ▶ Gathering roles used for your application using the Enterprise Archive (EAR) deployment descriptor.
- ▶ Configure authentication settings.
- ▶ Define resource reference.

These tasks must be completed for all of the applications participating in the SSO scenario. In this example, this means that both the Order Entry and Customer Inquiry applications must be configured, although the process is only described once, except when there are differences, and then both processes are described.

You will also need to import an external connector Resource Adaptor Archive (RAR) file into your WDSC projects.

7.1 Import an external Connector Resource Archive file into your project

Before you configure your applications to use EIM, you will need to import an external connector RAR file into your workspace inside WDSC. Here we describe the steps for WebSphere Development Studio Client for iSeries (WDSC) V6. If you are using WDSC iSeries V5.1.2, refer to online help.

Note: Two RAR files are provided:

- ▶ `eimIdTokenRA.rar` encapsulates `eim.jar` and facilitates deployment.
- ▶ `idTokenRA.rar` does not contain `eim.jar`, and requires the user to add it and configure the Server classpath.

1. From the Web or WebFacing Perspective, select your project and select **File** → **Import**.

2. Select **RAR file** and click **Next**, as shown in Figure 7-1.

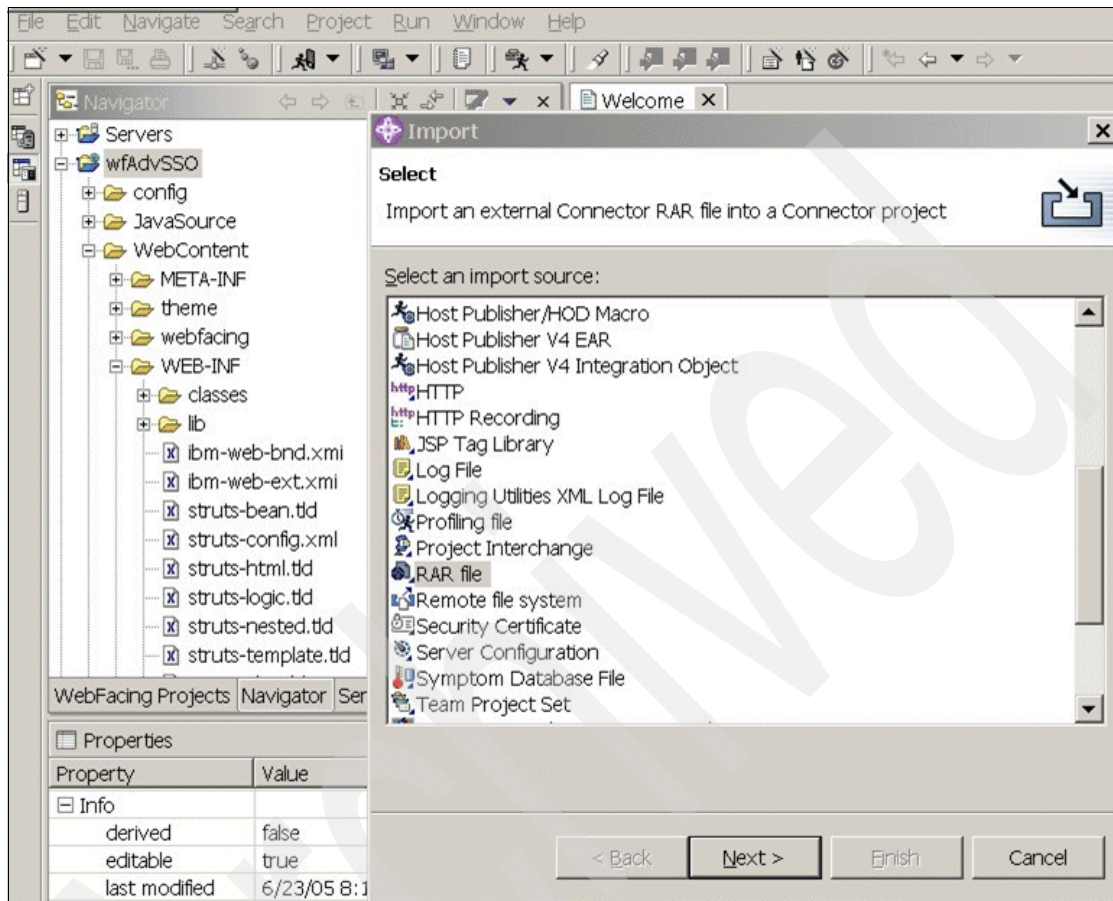


Figure 7-1 Import an external Connector RAR file

3. In the Connector Import dialog (see Figure 7-2 on page 147), select **Browse** and navigate to \Program Files\IBM\Rational\SDP\6.0\radi_prod\eclipse\plugins\com.ibm.etools.iseries.webtools.ae_6.0.1\lib. Select the eimIdTokenRA.rar file and click **Open** to add it to the Connector Import dialog.
4. Check **Add module to an EAR project**.
5. Select your EAR project from the list (in our sample, the project has the name wfAdvSSOEAR).
6. Click **Finish** to import the RAR file into the Web project.

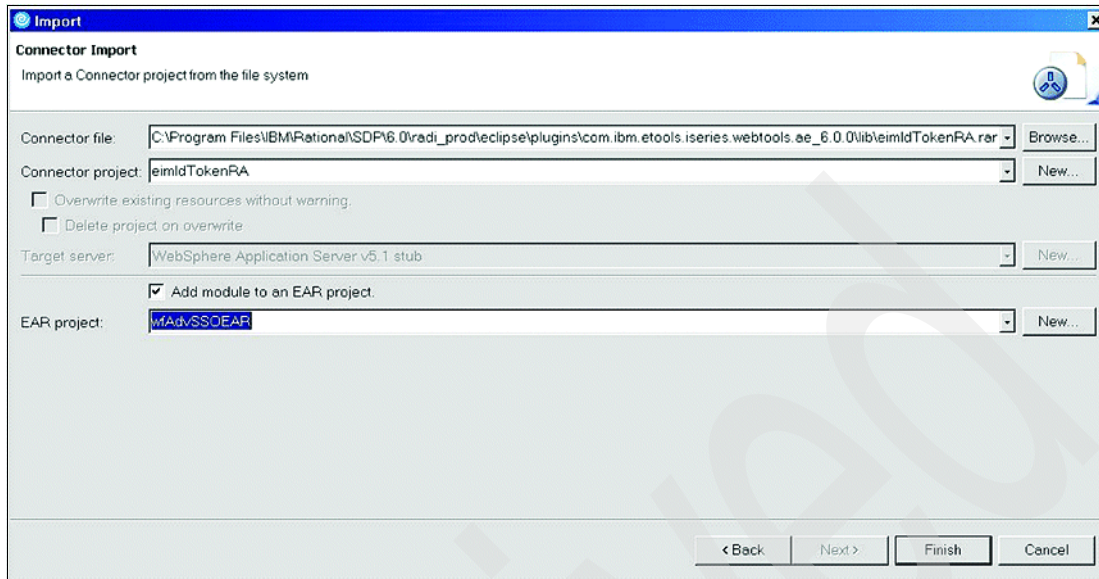


Figure 7-2 Connector Import

7.1.1 Configure the resource adapter to use the EIM domain

The steps to configure the resource adapter are done through the Application Deployment Descriptor:

1. To open the Application Deployment Descriptor editor, expand the EAR folder for your project in the Project Navigator and double-click the EAR Deployment Descriptor file `application.xml` under the `META-INF` folder.
2. Click the **Deployment** tab and expand the **Authentication** section.

3. Click **Add** to configure JAAS authentication for the LDAP administrator. Enter the user ID and password for the administrator and click **OK**. Figure 7-3 shows the Deployment Authentication page of the Application Deployment Descriptor.

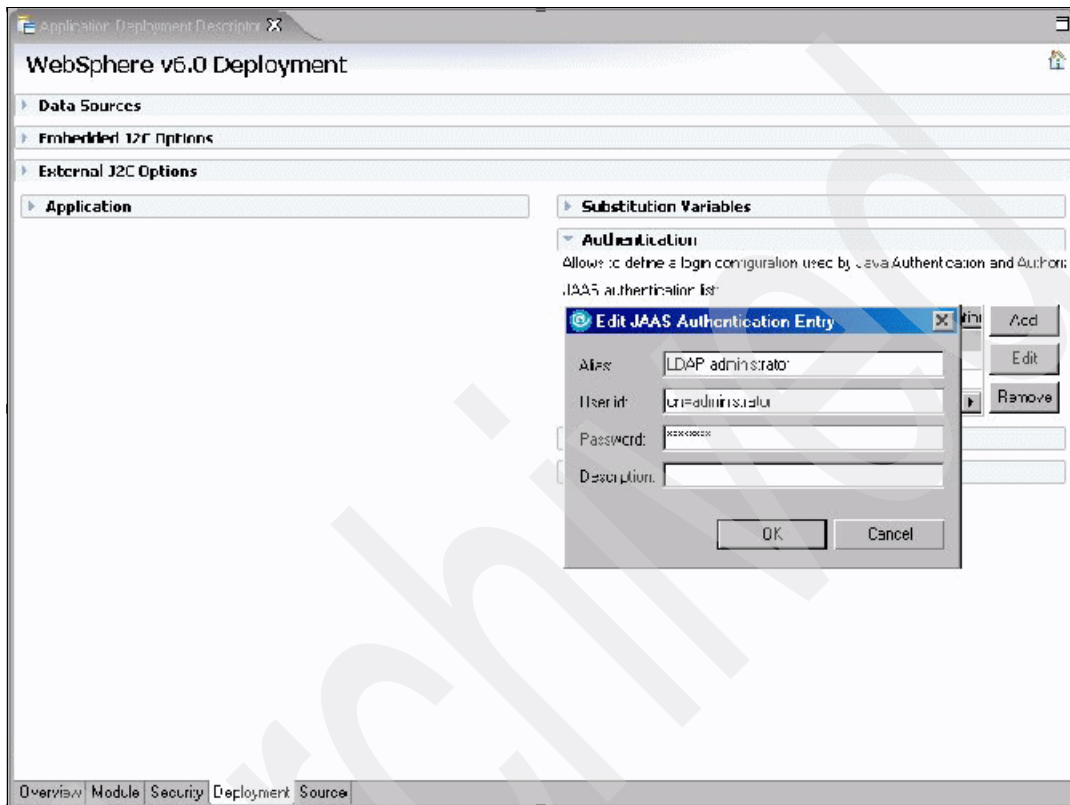


Figure 7-3 Application Deployment Descriptor - Deployment Authentication

4. In the Application Deployment Descriptor, expand **External J2C Options**.

5. Click **Add** next to J2C Resource Adapters. The Create Resource Adapter dialog is displayed with the resource adapter name `eimIdTokenRA` in the Resource Adapter Name field. Click **OK** to close the dialog. See Figure 7-4.

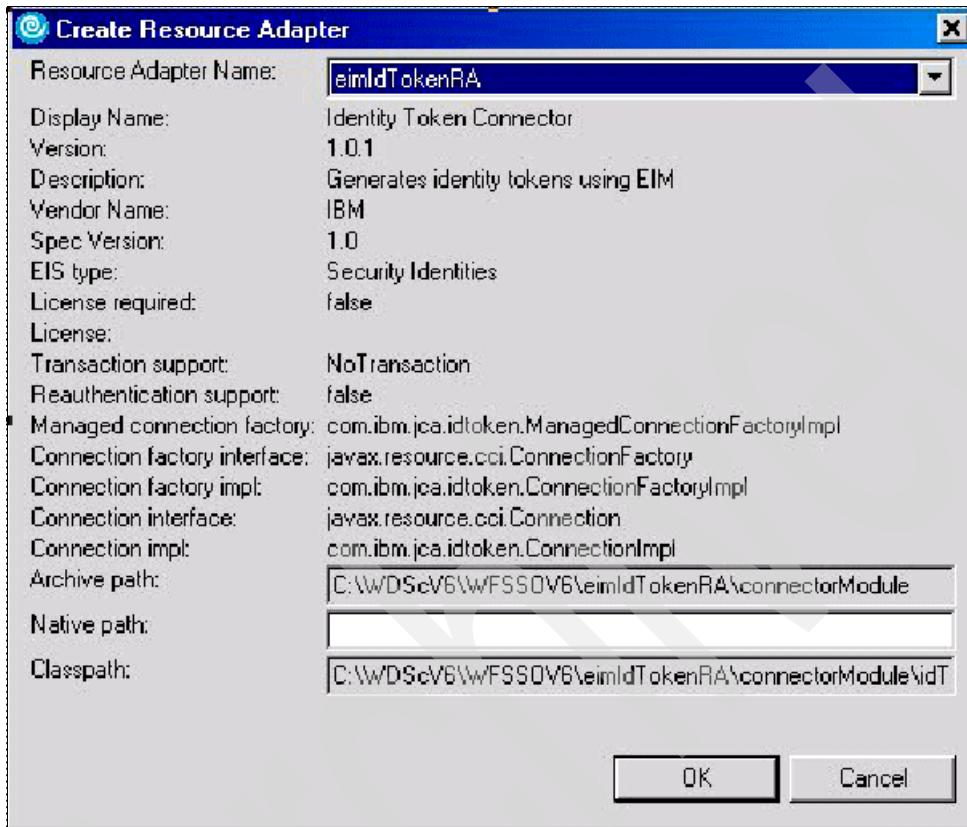
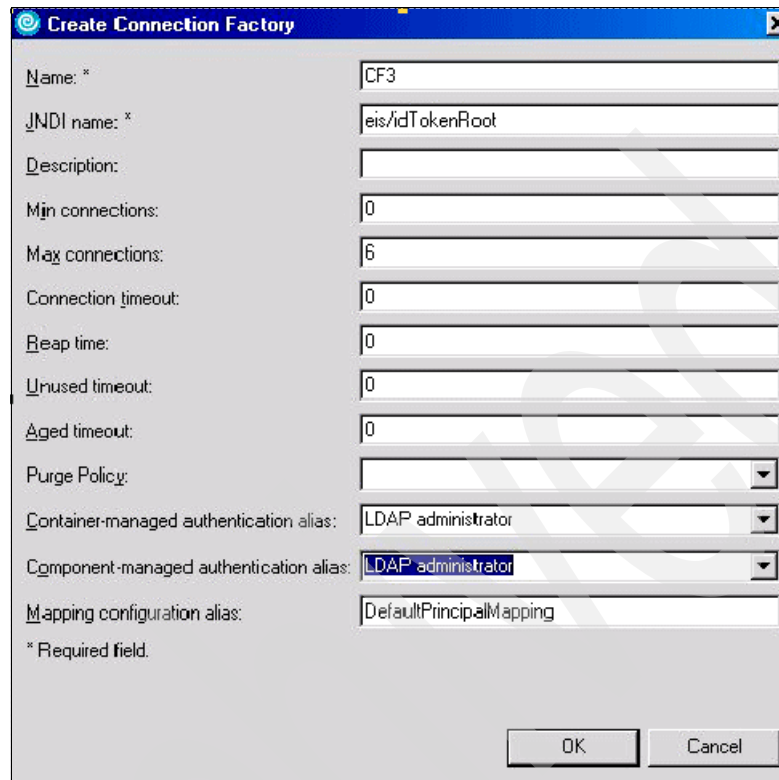


Figure 7-4 Create Resource Adapter

6. Click **Add** next to J2C Connection Factories. The Create Connection Factory dialog is displayed.
7. Enter a name for the connection factory; we use the name `CF3`.
8. Enter `eis/idTokenRoot` in the Java Naming and Directory Interface (JNDI) name field.
9. Set Max connections to 6.
10. Select the **LDAP administrator** for the Container-managed and Component-managed authentication aliases.
11. Leave all other fields with their default values and click **OK**.

Figure 7-5 shows the Create Connection Factory window.



The image shows a Windows-style dialog box titled "Create Connection Factory". It contains several fields for configuring a connection factory. The fields and their values are as follows:

Field	Value
Name: *	CF3
JNDI name: *	eis/idTokenRoot
Description:	
Min connections:	0
Max connections:	6
Connection timeout:	0
Reap time:	0
Unused timeout:	0
Aged timeout:	0
Purge Policy:	
Container-managed authentication alias:	LDAP administrator
Component-managed authentication alias:	LDAP administrator
Mapping configuration alias:	DefaultPrincipalMapping

At the bottom right, there are "OK" and "Cancel" buttons. A small asterisk (*) is followed by the text "Required field." at the bottom left.

Figure 7-5 Create Connection Factory

12. Under Resource Properties, provide the following values (specific to your location):
 - a. LDAPHostName (in our scenario, as270dd.duedorf.de.ibm.com).
 - b. EimDomainName: Enter the name of your EIM domain, specified during EIM configuration, in our sample, eim_ffts.
 - c. ParentDomain: Enter the base distinguished name for the EIM domain. This value was also specified during EIM configuration, in our sample, dc=as270dd,dc=duedorf,dc=de,dc=ibm,dc=com.
 - d. SourceRegistryName is the name of the EIM registry in which the authenticated user name has a source mapping, In our sample, WebSphere Registry.

Figure 7-6 shows the Resource Adapter properties in WDSC.

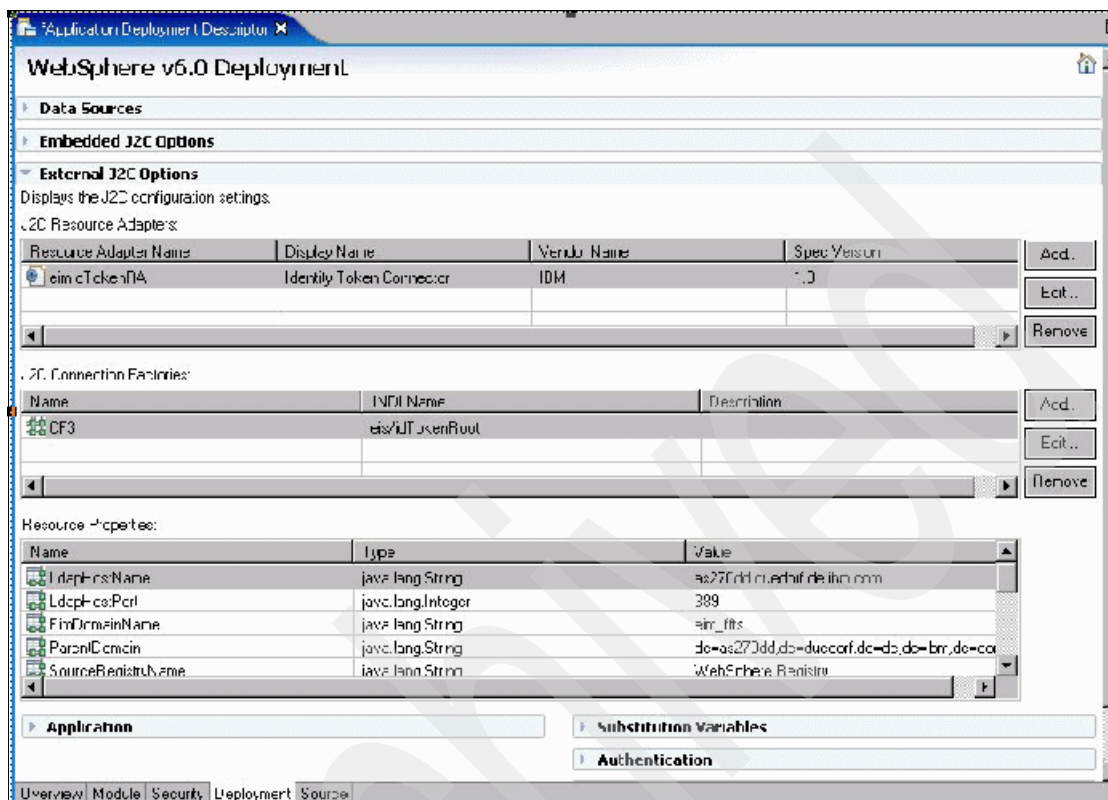


Figure 7-6 Resource Adapter properties in WDSC

13. Close and save the Application Deployment Descriptor.

7.2 Setting up security roles and constraints for your application

A security role is a logical grouping of users (for example, Bank Teller and Bank Manager). A security constraint defines which part of the application to secure (for example, servlet or JavaServer™ Page components) and which security roles can access them.

Security roles and constraint are defined in the Web Deployment Descriptor (web.xml) file inside your WDSC project. The Security window has two sections:

1. Security Roles, which display the security roles defined for this Web application as well as descriptions of each role. Roles can be added, removed, and edited here as well.
2. Security Constraints, which supports adding or removing security constraints for specific security roles as well as adding descriptions of each security constraint.

7.2.1 Define a security role

Let us now define the security roles and constraints for the WebFacing (Order Entry) application.

To define security roles and constraints, perform the following steps:

1. Open the Project Navigator (Navigator tab), as shown in Figure 7-7 on page 153.
2. Expand the Web project folder of your WebFacing Order Entry Application (in our case, the name is wfAdvSSO) and then the **WebContent** → **WEB-INF** folder.
3. Double-click **web.xml** to open the Web Deployment Descriptor in the editor.
4. Click the **Security** tab on the bottom of the window.
5. Click the **Add** button under the Security Roles section to insert a new security role.
6. This example uses the role All application users to specify all users in the LDAP directory on the iSeries server. Therefore, the name All application users is appropriate for the Name field of the Security Role window. Click **Finish**.

Figure 7-7 shows the web.xml editor in the Security Role window.

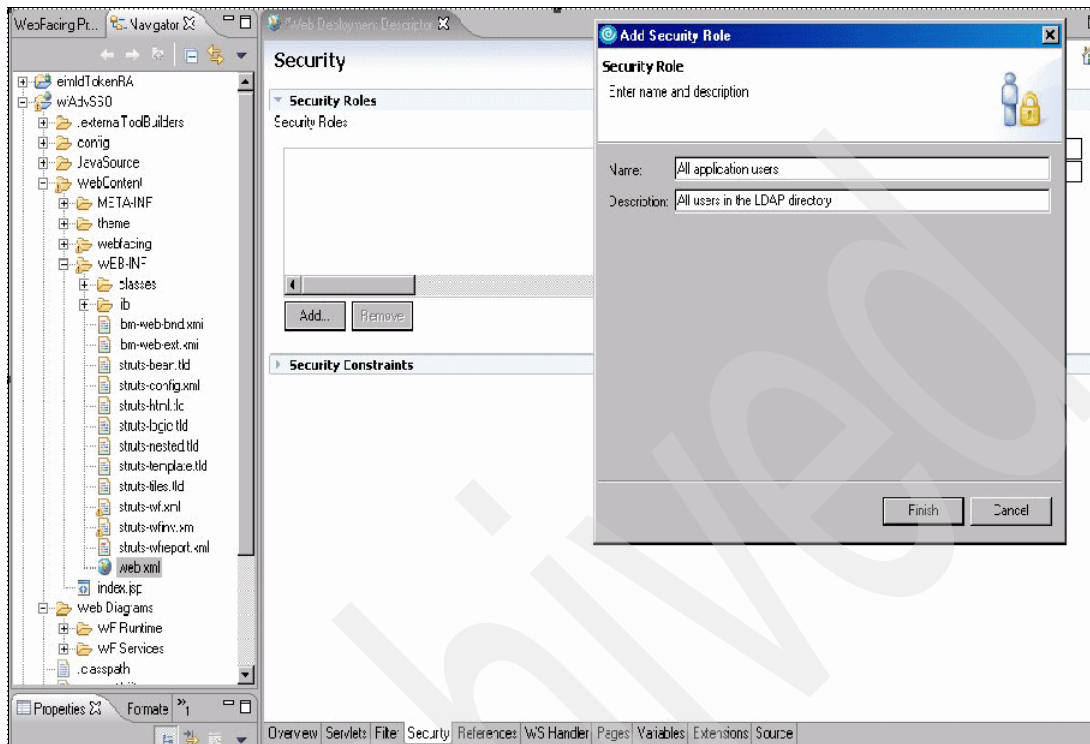


Figure 7-7 Security Role, web.xml editor

7.2.2 Define a security constraint

To define a security constraint, do the following steps:

1. We assume you are still on the Security tab site of the web.xml (Web Deployment Descriptor) editor. If not, repeat the described steps 1 on page 152 to 4 on page 152.

2. In the Security Constraints section, click the **Add** button, to define the security constraints for this new security role. Type the Constraint name All user constraint and click **Next**. See Figure 7-8.

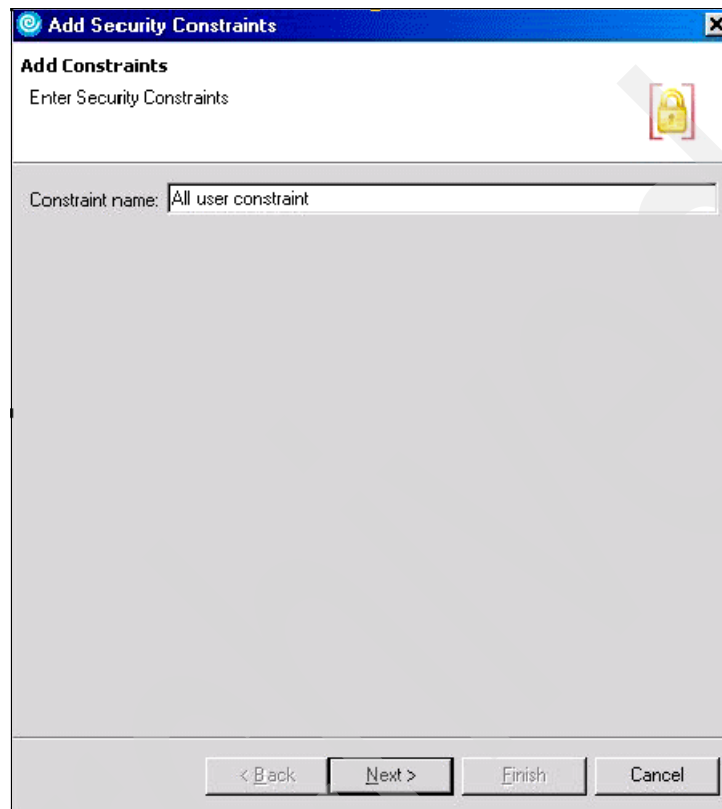


Figure 7-8 Add Security Constraints

3. The Add Web Resource window comes up. Here we associate a set of Web resources with this security constraint for this security role. A Web resource collection defines a set of URL patterns (resources) and HTTP methods belonging to the resource. HTTP methods handle HTTP requests, such as GET, POST, PUT, and DELETE.
 - e. Type GetAndPost as the name for this set of constraints.
 - f. Select the **Get** and **Post** check boxes. By doing so, this security role has the authority only to perform get and post methods for the application. If no HTTP methods are specified, then the security constraint applies to all HTTP methods.

- g. Beside the Pattern field, click the **Add** button and enter `/*` for the URL pattern. This entry specifies that the security role has access to all URL patterns.

A URL pattern specifies which levels of the URL can be accessed by the security role. For example, a Web application may contain many resources or windows stored in subdirectories that are separated by a slash in the URL. For the URL `http://www.myCompany.com/customers/address.html`, customer and address resources are stored in a different directory than the Order Entry application. The URL pattern specified on this dialog box denotes which level can be accessed by the security role.

Click **OK** to add this definition. See Figure 7-9.

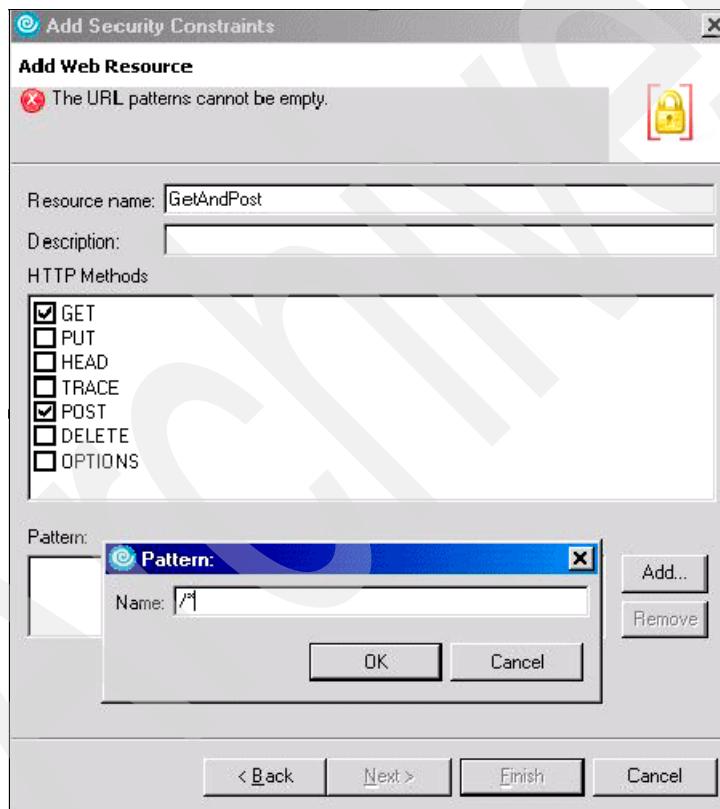


Figure 7-9 Security Constraints - Add Web Resources

4. In the Add Web Resource window, click **Finish**.
5. Back in the Security part of the Deployment Descriptor file you see the Security Constraint and Web Resource Collection you just created.

Figure 7-10 shows the Security part of the Web Deployment Descriptor.

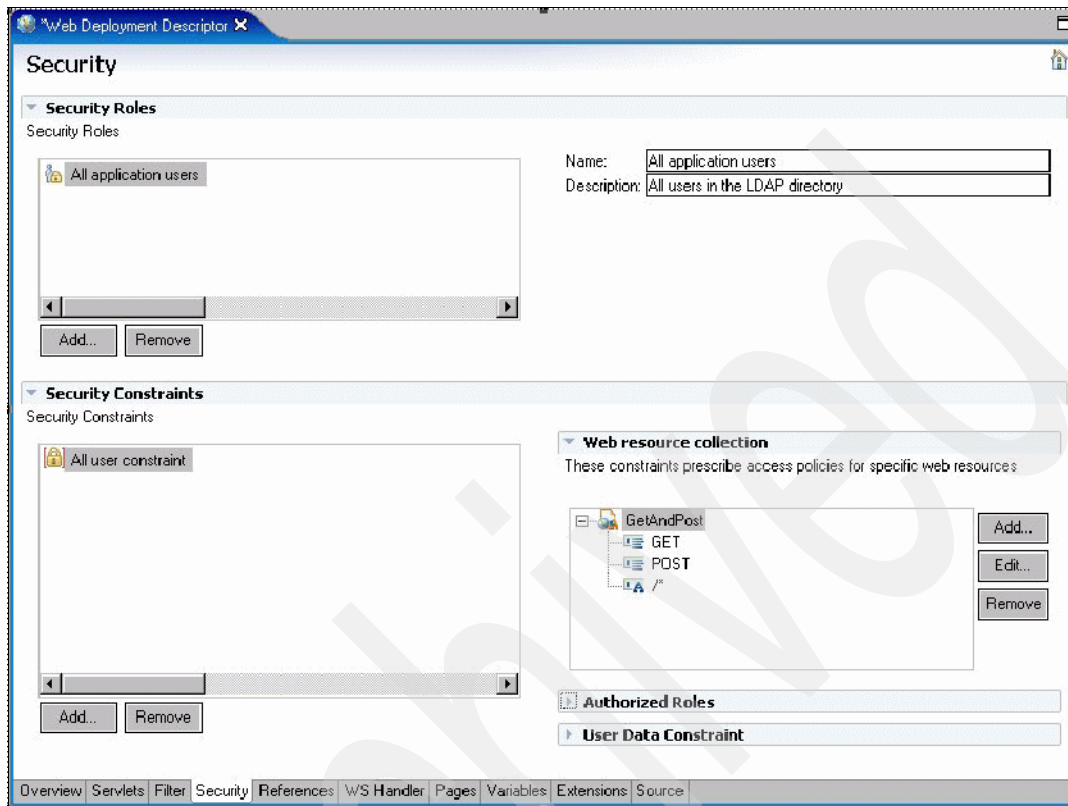


Figure 7-10 Security part of the Web Deployment Descriptor

6. Click the **Add** button in the Authorized Roles section to open the Define Authorization Constraint window.

Check **All application users** to associate this role with the Web resource collection you just defined and click **Finish**. See Figure 7-11.

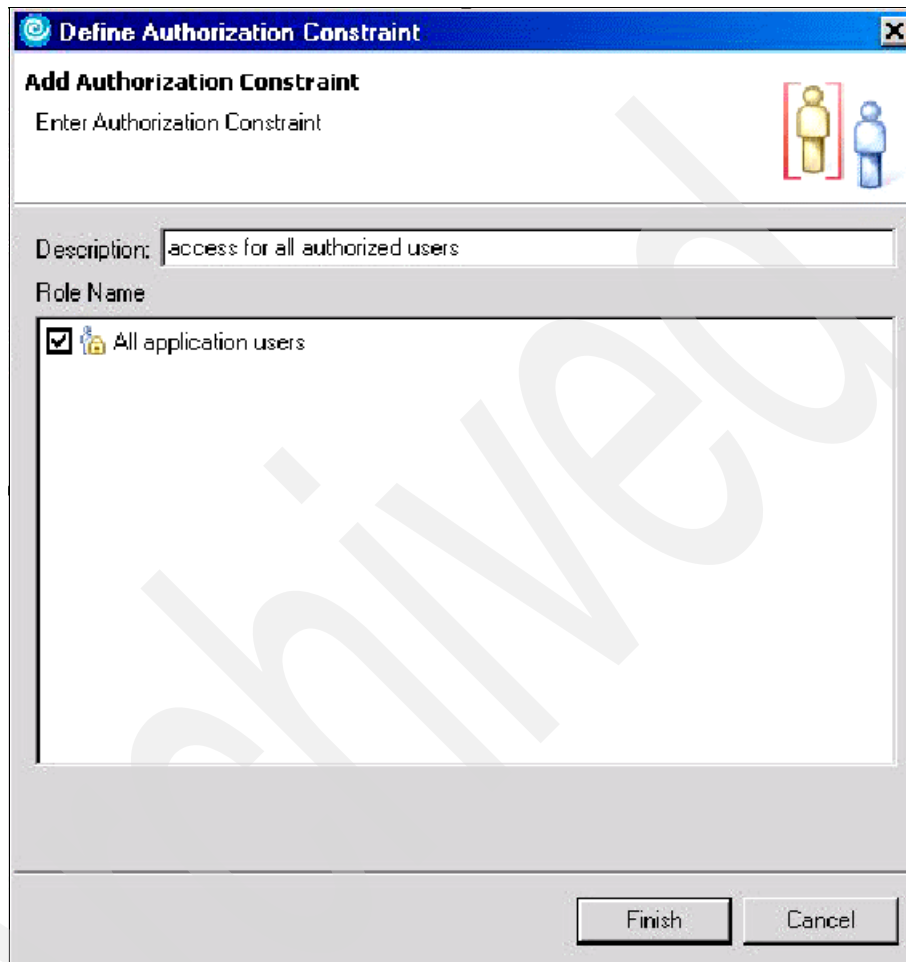


Figure 7-11 Add Authorization Constraint

7. The final definitions in the Security part of the Deployment Descriptor file should be similar to those shown in Figure 7-12. Close and save the Web Deployment Descriptor file web.xml.

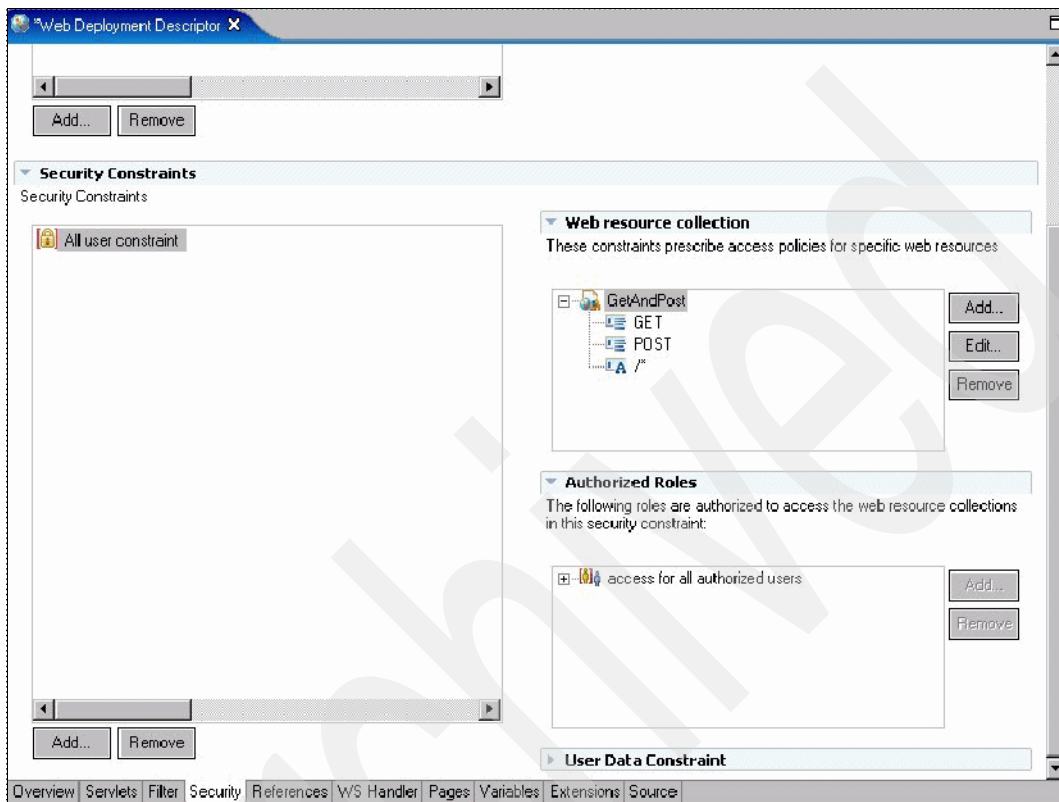


Figure 7-12 Security Constraints for application - final

7.3 Consolidating security roles

You may find that, for an enterprise application, there will be some overlap in the security roles that are defined in the application's modules. If so, you can combine and remove redundant or unnecessary security roles by using the Gather function on the Security tab in Project Navigator.

To associate application EAR file with roles and constraints, do the following:

1. In the Project Navigator, expand the EAR folder for your project (wfAdvSSOEAR in our case) and double-click the EAR Deployment

Descriptor file application.xml to open it in the editor. Click the **Security** tab and then click the **Gather** button.

2. Select the just gathered security role **All application users** on the left site of the window. Check **All authenticated users** under WebSphere Bindings. Binding information is required by the application server to bind the deployment information specified in the application to a specific instance. In this example, it is mapping a security role, all application users, to a set of groups or users registered on WebSphere Application Server, which in this case is all authenticated users. See Figure 7-13.

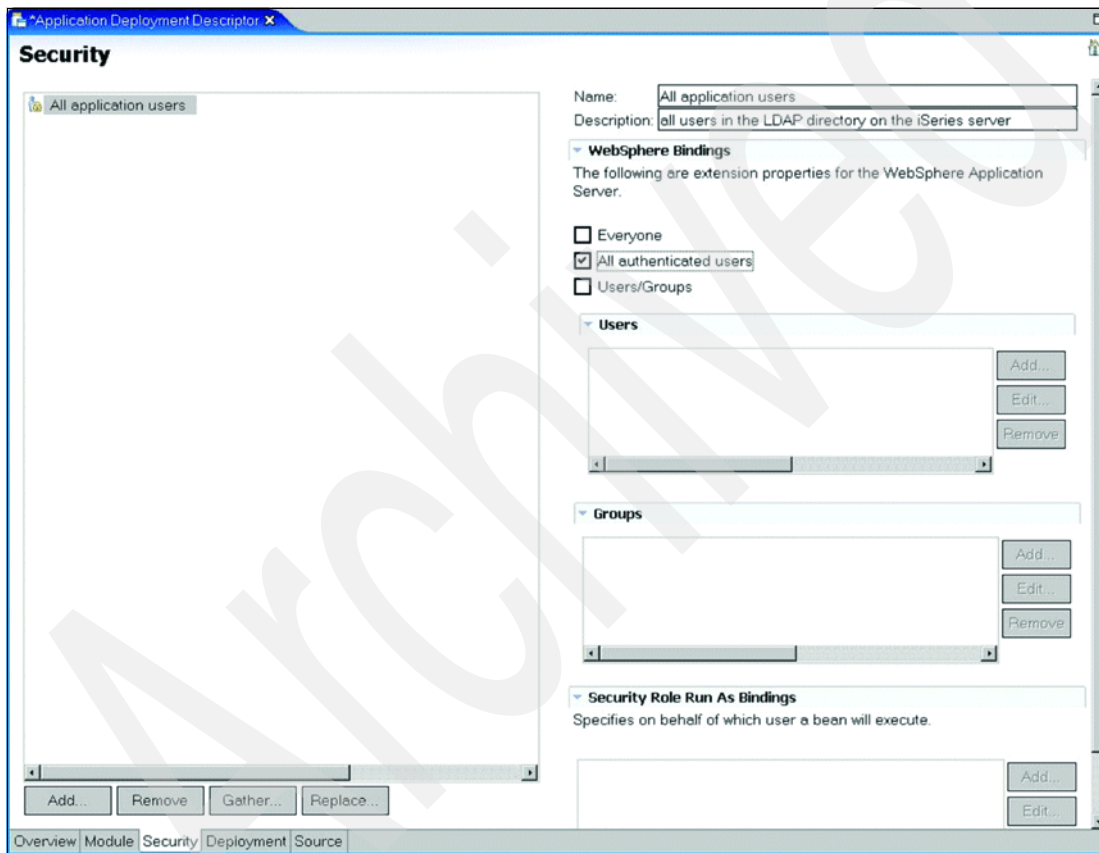


Figure 7-13 Application Deployment Descriptor - application.xml file - Security tap

3. Close and save the Application Deployment Descriptor file.

4. Now that security has been enabled for the WebFacing application, the Customer Inquiry Web application must also be secured. Repeat the steps in “Setting up security roles and constraints for your application” on page 151 and “Consolidating security roles” on page 158 for the Customer Inquiry application.

7.4 Configure authentication settings

You have to configure authentication settings for both applications. Since the Order Entry application was created using the WebFacing tool, and the Customer Inquiry using the Web Tools, the steps to configure the authentication settings are slightly different. For this reason, we will step through both processes individually.

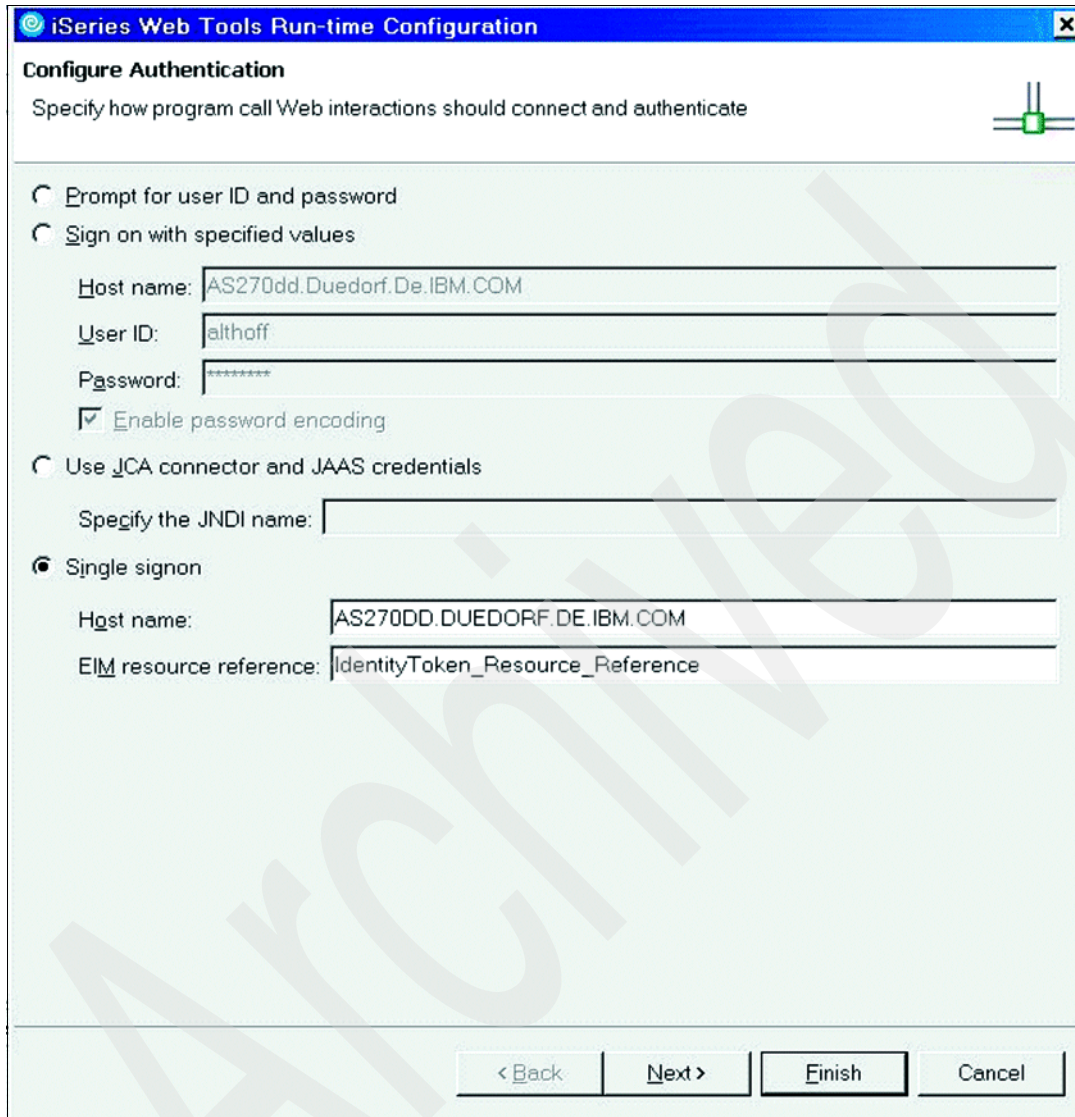
7.4.1 Configure the authentication settings for the Customer Inquiry application

You will use the iSeries Web Tools Run-time Configuration dialog in WDSC (see Figure 7-14 on page 161) to configure the authentication settings for the Customer Inquiry application.

1. In the Web perspective, right-click the project and click **Specify the iSeries Web Tools Run-time Configuration**.
2. Select the **Single signon** radio button.
3. Type the iSeries host name in the host name field. Our scenario uses AS270DD.DUEDORF.DE.IBM.COM.
4. Type IdentityToken_Resource_Reference in the EIM resource reference field. Click **Finish**.

The previous steps mapped the token name used by the application to the previously configured RAR file. The name itself is just a string and so can be given any value.

In 5.2, “Components needed for SSO” on page 95, the create WebSphere Application Server wizard created the connection factory CF1 and assigned the JNDI name as “eis/IdentityToken”, where “eis” denotes that it is an enterprise-information system and IdentityToken is the arbitrary string name. Since that name needs to be mapped by being mapped to, we gave it the string IdentityToken_Resource_Reference. By specifying IdentityToken_Resource_Reference as the EIM resource, the application will use the resource adapter that was created by the WebSphere Application Server wizard.



The image shows a Windows-style dialog box titled "iSeries Web Tools Run-time Configuration" with a sub-header "Configure Authentication". The main instruction is "Specify how program call Web interactions should connect and authenticate". There are three radio button options: "Prompt for user ID and password", "Sign on with specified values", and "Use JCA connector and JAAS credentials". The "Sign on with specified values" option is selected. Below it are text fields for "Host name" (AS270dd.Duedorf.De.IBM.COM), "User ID" (althoff), and "Password" (masked with asterisks). A checkbox "Enable password encoding" is checked. The "Use JCA connector and JAAS credentials" option is unselected, with a "Specify the JNDI name:" label and an empty text field. The "Single signon" option is selected. Below it are text fields for "Host name" (AS270DD.DUEDORF.DE.IBM.COM) and "EIM resource reference" (IdentityToken_Resource_Reference). At the bottom are four buttons: "< Back", "Next >", "Finish", and "Cancel".

iSeries Web Tools Run-time Configuration

Configure Authentication

Specify how program call Web interactions should connect and authenticate

☐ Prompt for user ID and password

☒ Sign on with specified values

Host name: AS270dd.Duedorf.De.IBM.COM

User ID: althoff

Password: *****

☒ Enable password encoding

☐ Use JCA connector and JAAS credentials

Specify the JNDI name:

☒ Single signon

Host name: AS270DD.DUEDORF.DE.IBM.COM

EIM resource reference: IdentityToken_Resource_Reference

< Back Next > Finish Cancel

Figure 7-14 Web Tools Run-time Configuration-Configure Authentication

7.4.2 Configure Authentication settings for the Order Entry application (WebFacing)

Next, you need to configure authentication settings for the Order Entry application. Since you are already using the WebFacing Tool, you can configure authentication settings in the Run Time properties for your project inside WDSC.

1. Open the WebFacing perspective. By default, the upper left pane contains WebFacing and Resource Navigator configuration views. Make sure the WebFacing view is active.
2. Right-click the project and select **Properties**.
3. In the Properties window (Figure 7-15 on page 163), expand the tree branch under **Run Time** → **Project**.
4. Open the **Authentication** tab.
5. Select the **Single signon** radio button.

Note: Selecting single sign-on disables other authentication options on the Authentication settings window.

6. Check the **Specify EIM resource reference** and enter IdentityToken_Resource_Reference. Click **OK**.

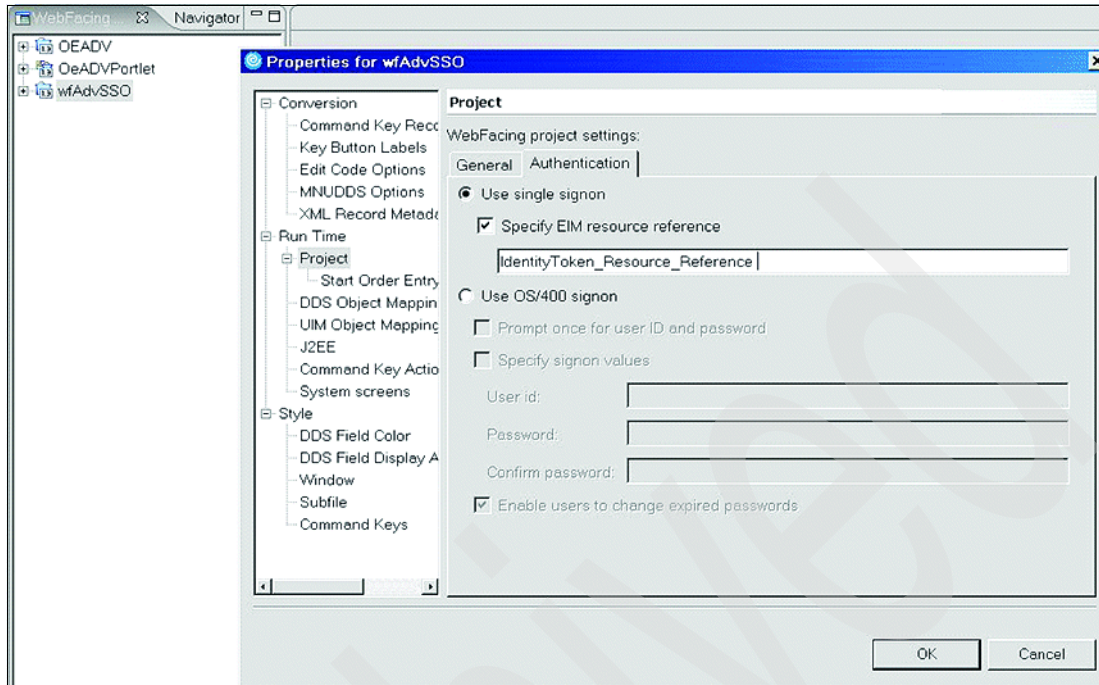


Figure 7-15 WebFacing project settings - Authentication

7.5 Define resource reference for both applications

Next we will define the resource references. In this section, we describe the steps taken for the Order Entry application. You will use the same procedure to define the resource references for the Customer Inquiry application.

Open the Resource Navigator (not the WebFacing view) and:

1. Expand the Web project folder **wfAdvSSO** → **WebContent** → **WEB-INF** and then double-click **web.xml** to open it in the editor.
2. Click the **References** tab to configure the resource reference for the application.
3. In section References in the left side of the window, click the **Add** button.

4. In the Add Reference window, select **Resource reference** and click **Next** (Figure 7-16).

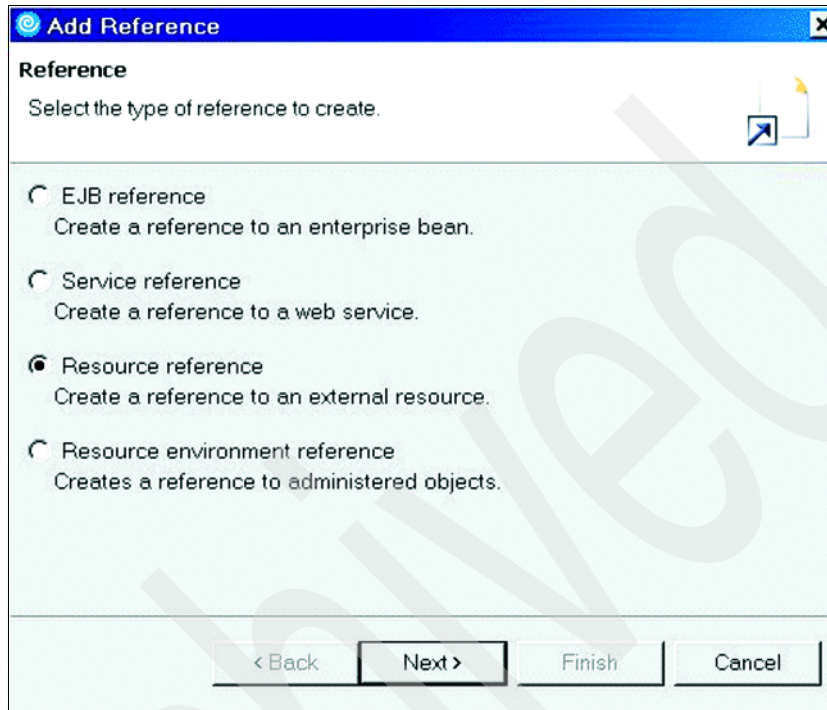
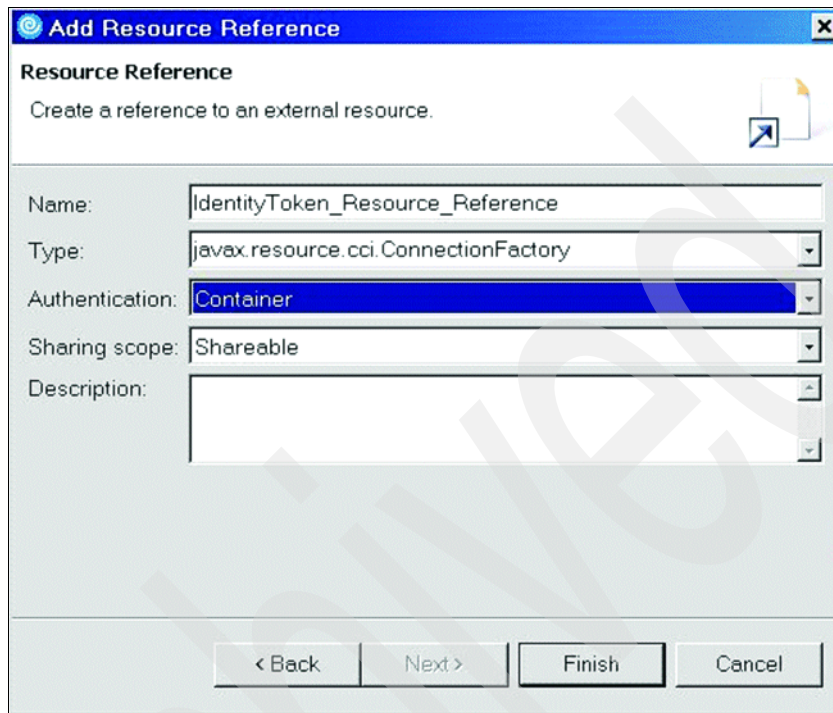


Figure 7-16 Add Reference - select Resource reference

5. Type `IdentityToken_Resource_Reference` in the Name field. This is the same value used in Figure 7-15 on page 163.
6. Select `javax.resource.cciConnectionFactory` from the Type selection list. If it is not shown in the selection list, type this value in.
7. Select **Container** from the Authentication selection list.

8. Select **Sharable** from the Sharing scope selection list. Click **Finish**. See Figure 7-17.



The image shows a Windows-style dialog box titled "Add Resource Reference". Below the title bar, the text "Resource Reference" is displayed, followed by the instruction "Create a reference to an external resource." and a small icon of a document with a blue arrow. The dialog contains several input fields: "Name:" with the text "IdentityToken_Resource_Reference"; "Type:" with a dropdown menu showing "javax.resource.cci.ConnectionFactory"; "Authentication:" with a dropdown menu showing "Container"; "Sharing scope:" with a dropdown menu showing "Shareable"; and "Description:" with an empty text area. At the bottom, there are four buttons: "< Back", "Next >", "Finish", and "Cancel".

Figure 7-17 Define Resource Reference

9. Back in the References window, enter `eis/idTokenRoot` in the JNDI Name field under WebSphere Bindings. Just as a refresher, this was the JNDI name we specified when configuring the connection factory. For the relationship, see Figure 7-19 on page 167.

Figure 7-18 shows the Final resource Reference in Web Deployment Descriptor.

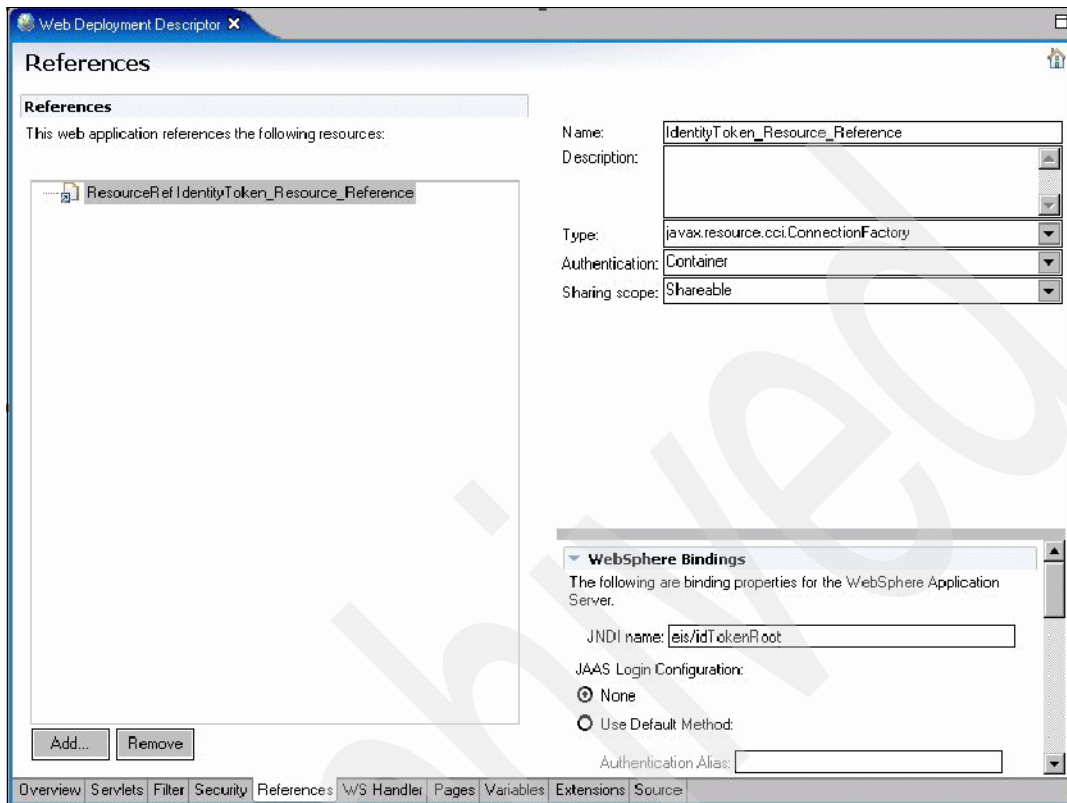


Figure 7-18 Final Resource Reference in Web Deployment Descriptor

Figure 7-19 shows the Connection Factory and Resource Reference.

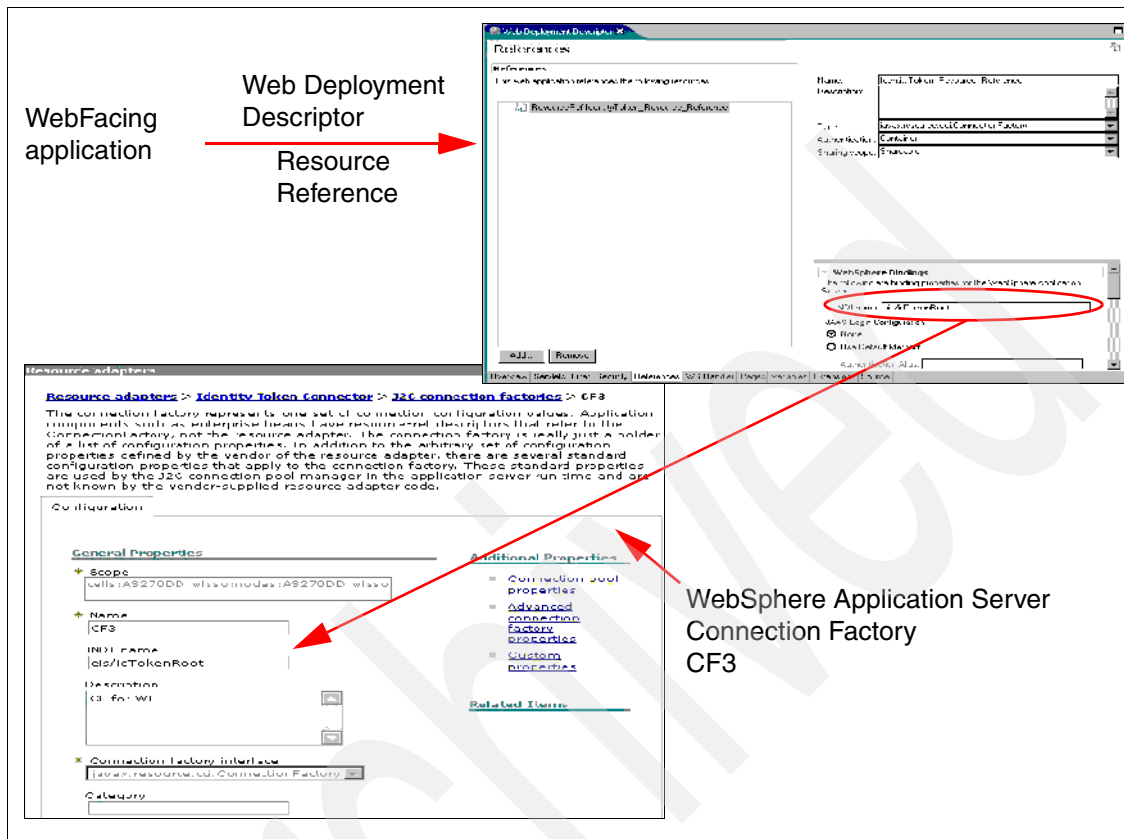


Figure 7-19 Connection Factory and Resource Reference

10. Save and close the Web Deployment Descriptor file.

7.6 Implement link to the Order Entry application

In order to test our SSO implementation, one application needs to call the other. If the configuration is a success, the user should only have to log on once even though the user is moving between applications. We will define a hyperlink on the Customer Inquiry application input.jsp file to call the Order Entry application.

Perform the following steps:

1. In WDSC, switch to the Web perspective.
2. Expand the **Customer Inquiry project** folder and the **Web Content** folder.

3. Double-click the result.jsp file to open it in the editor window.
4. Switch to the Design view by clicking the **Design** tab at the bottom of the editor window.
5. Right-click anywhere in the editor window and select **Insert Link**. The Insert Link window opens.
6. Select the **File** radio button to specify file as the type of link.
7. Click **Browse** to choose the target URL, which in this case is the Order Entry index.jsp file.
8. Select the index.jsp file and click **OK** to close the browse window.
9. The URL and Link text fields should now be populated with the location of the project you just selected.
10. Type Launch the Order Entry Application in the Link text field. This is the hyperlink text that will appear on the Customer Inquiry's index window. The Target field can be left blank so that, by default, the target application will open in the same window as the first application. You can also have it open in a new window.
11. Click **OK** to close the window and insert the hyperlink on the inquiry.jsp window.
12. Save the changes.

The applications are now ready to run, and the next step is to export the applications from WDSC, before both applications are published on the server.

7.7 Export the applications from WDSC

After completing the configuration steps for your applications as described in this chapter, you are ready to export these applications from WDSC.

1. From the Resource Navigator in WDSC, select the project that you wish to export, right-click it, and select **Export**.
2. From the Export window, select **EAR file** and click **Next**.
3. In the EAR project selection list, select the project that you will export.

Beside the Destination, use the **Browse** button to select the directory where you will export the EAR file. We will do that through the mapped network drive of our System i and export the EAR file of our project to directory Z:\QIBM\UserData\WebSphere\AppServer\V6\Base\profiles\wfsso\installable Apps. Click **Finish**. See Figure 7-20 on page 169.

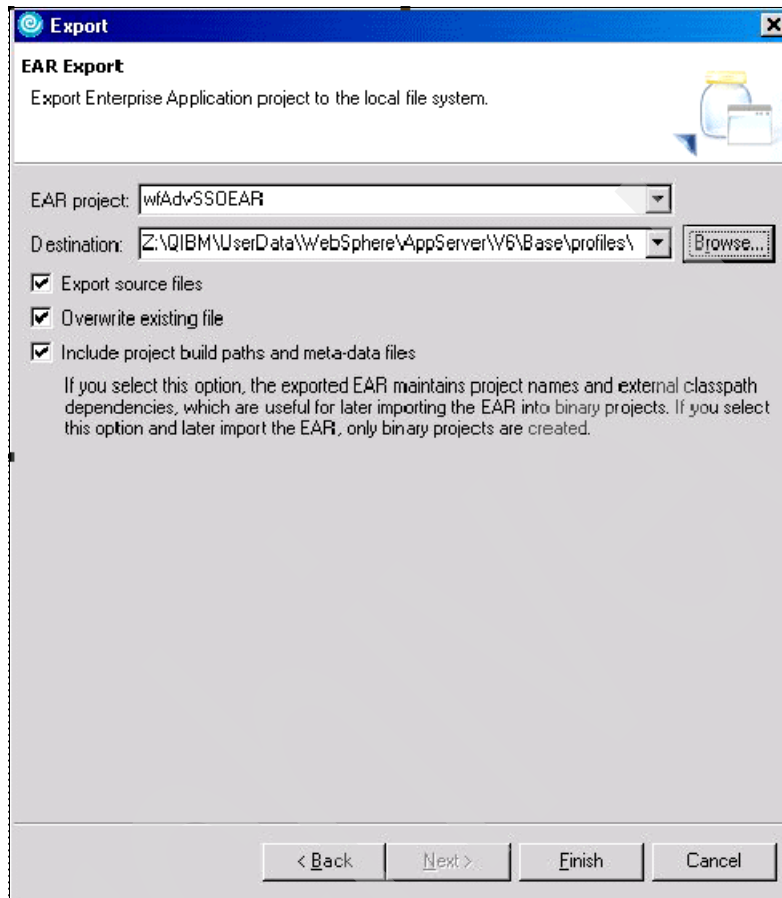


Figure 7-20 Export Project from WDSC

The SSO sample application

There is a sample application available that can be used for testing our SSO environment. This sample application uses an EIM Identity Token connection factory to provide EIM identity tokens for use with IBM Toolbox for Java `com.ibm.as400.access.AS400` objects. For example, if the sample application is deployed on SYSTEM A, you can log in just once to the WebSphere Application Server and use the sample application to perform OS/400 system commands under your OS/400 user profiles on SYSTEM B, SYSTEM C, or SYSTEM D.

8.1 How it works

When you make a request to the sample application, you must log in with your WebSphere user ID and password. Each request contains the system command and the target system name where the command is to be executed. When the request is received, the application calls the connection factory to generate an identity token. The connection factory extracts your user ID from a Java Authentication and Authorization Service (JAAS) Subject object provided by WebSphere security and collaborates with the EIM Domain Controller to create the identity token that it returns to the application. The application then creates a `com.ibm.as400.access.AS400` object for SYSTEM B, providing it with the identity token instead of your OS/400 user profile before passing it the system command to be executed.

Note: A new identity token and AS400 object is created each time you send a request that contains a new target system, and all AS400 objects are stored in an HTTP Session for use with subsequent requests.

8.2 Installing the sample application

1. From the WebSphere Administrative Console, expand **Applications** and click **Install New Applications**.
2. Select **Local Path** if you have mapped a drive to your iSeries system; otherwise, select **Server path**.
3. Type or browse to provide the path name for the EAR file `testidentitytoken.ear` (that is, `/QIBM/ProdData/OS400/Java400/security/eim/testidentitytoken.ear` on your System i5 V5R3 system). Click **Next**.
4. Check **Use default virtual host name** under Virtual Host, and then click **Next**.
5. Ignore the warning similar to that shown in Figure 8-1 on page 173 and click **Continue**.

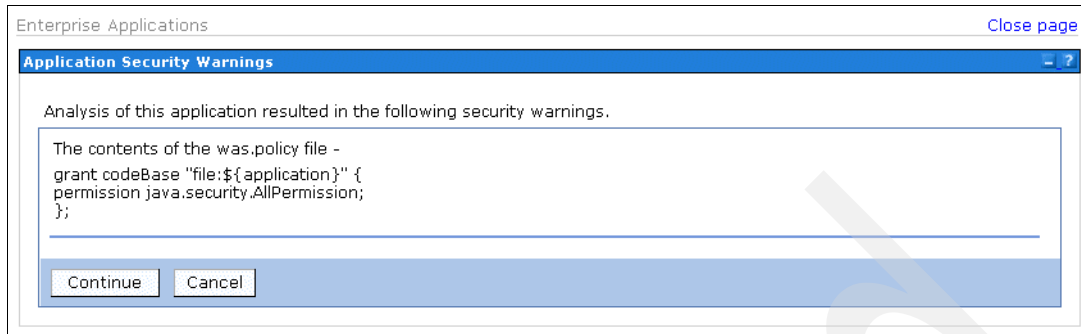


Figure 8-1 Application Security Warnings

6. In Step 1 leave all the parameters as the default and then select the **Installation Options** window and click **Next**.
7. In Step 2, in the Map modules to servers window, select the **testIdentityTokenWeb** module and mark both servers as shown in the Cluster and Servers selection list and click **Apply**.

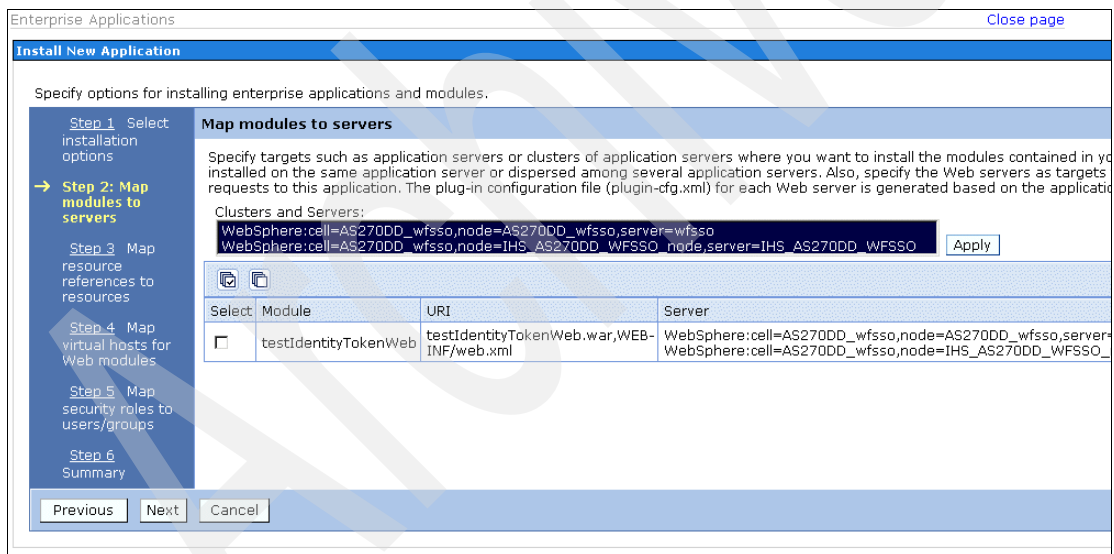


Figure 8-2 Install Application - Step 2

8. You should see that both servers are added under the Server list for the Module testIdentityTokenWeb, as shown in Figure 8-3.

Server
WebSphere:cell=AS270DD_wfsso,node=AS270DD_wfsso,server=wfsso
WebSphere:cell=AS270DD_wfsso,node=IHS_AS270DD_WFSSO_node,server=IHS_AS270DD_WFSSO

Figure 8-3 Install Application - Step 2 with mapped servers

9. Step 3, the Map resource references to resources window, allows you to change the Java Naming and Directory Interface (JNDI) Name for Reference Binding eis/IdentityToken_Shared_Reference. Do this only if you have configured your connection factory with a JNDI Name other than eis/IdentityToken. Click **Next**.
10. Ignore the warning, Application Resource Warnings when it comes up. Click **Continue**.
11. In Step 4, the Map virtual hosts for Web modules window, leave the Virtual host to value default-host and click **Next**.
12. Step 5 is the Map security roles to users/groups window shown in Figure 8-4. Leave the **All authenticated** option checked and click **Next**.

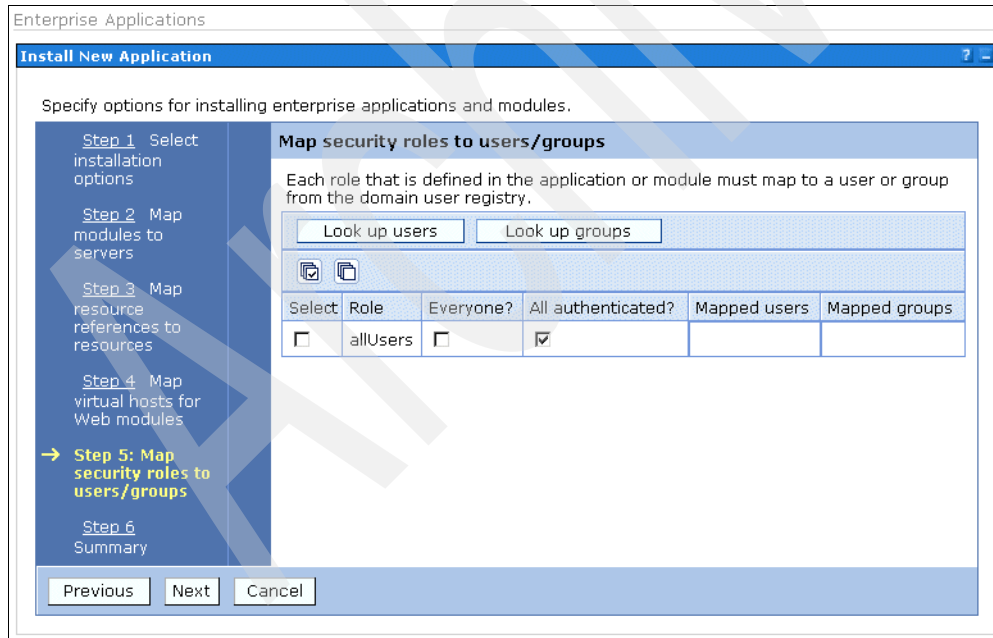


Figure 8-4 Install Application - Step 5 with mapped security roles

13. When you get to Step 6, Summary, just click the **Finish** button.
14. On the Installation window, look for the message “Application testIdentityToken installed successfully” and click **Save to Master Configuration**.
15. Save the changes.

8.3 Start the sample application

Before you can test the newly installed application, you need to start the application using the following steps:

1. From the WebSphere Administrative Console, expand **Applications** and click **Enterprise Applications**.
2. Check the testIdentityToken application and click **Start**. Figure 8-5 shows the Start the new installed application window.

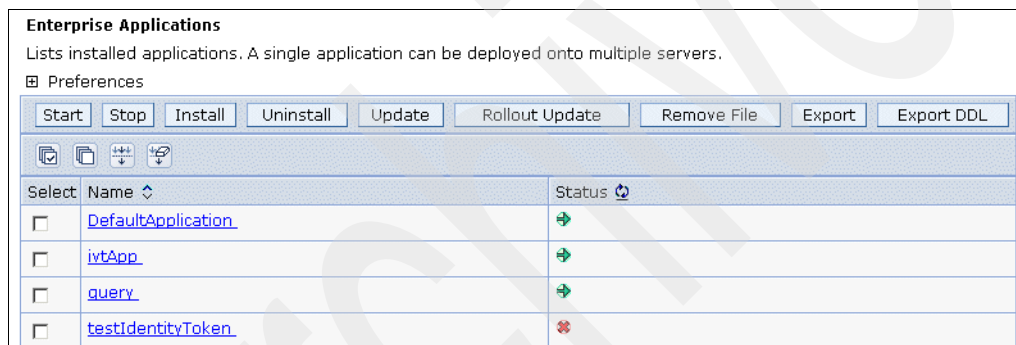


Figure 8-5 Start the new installed application

3. In the Status list you will see than a green arrow. which means that the application is started successfully.

8.4 Test the Identity Token sample application

Now you are ready to test the Identity Token sample application.

1. Open a new session of your Web browser.
2. Request the application's welcome page from your Web browser using URL `http://hostname_and_port/testIdentityTokenWeb/IDTknTest.jsp`, where `hostname_and_port` is the host name and port for your external Web server or internal (WebSphere container) HTTP transport.

In our environment it is:

`http://as270dd.duedorf.de.ibm.com:40400/testIdentityTokenWeb/IDTknTest.jsp`

3. Type a value for the OS/400 Host System Name and for the OS/400 Command. For example, if you have EIM configured for system `mssystem`, then type `mssystem` in the OS/400 Host System Name field and `crtlib mylib` in the OS/400 Command field.

In our case, we type the values `as270dd.duedorf.de.ibm.com` for our System `i5` and `crtlib test1` as the OS/400 Command, as shown in Figure 8-6.

Identity Token Test Client

This test page is used to test the identity token support by invoking a servlet that then uses the identity token connector to generate an identity token for use with the IBM Java toolbox.

Enter the information required to use the Java toolbox, then click Submit. The results of the OS/400 CL command will be displayed.

OS/400 Host System Name	as270dd.duedorf.de.ibm.com
OS/400 Command	crtlib test1
Cache Reset	<input type="checkbox"/>

Figure 8-6 EIM Identity Token sample application

4. Click **Submit**.
5. Type a user ID and password on the login prompt that comes up. Use a user that is already defined in the LDAP directory and has also defined an EIM identifier to map the i5/OS user profile.

In our scenario, we use the user Ursula Althoff with the according password, which has the EIM associations, as shown in the Figure 8-7 on page 177.

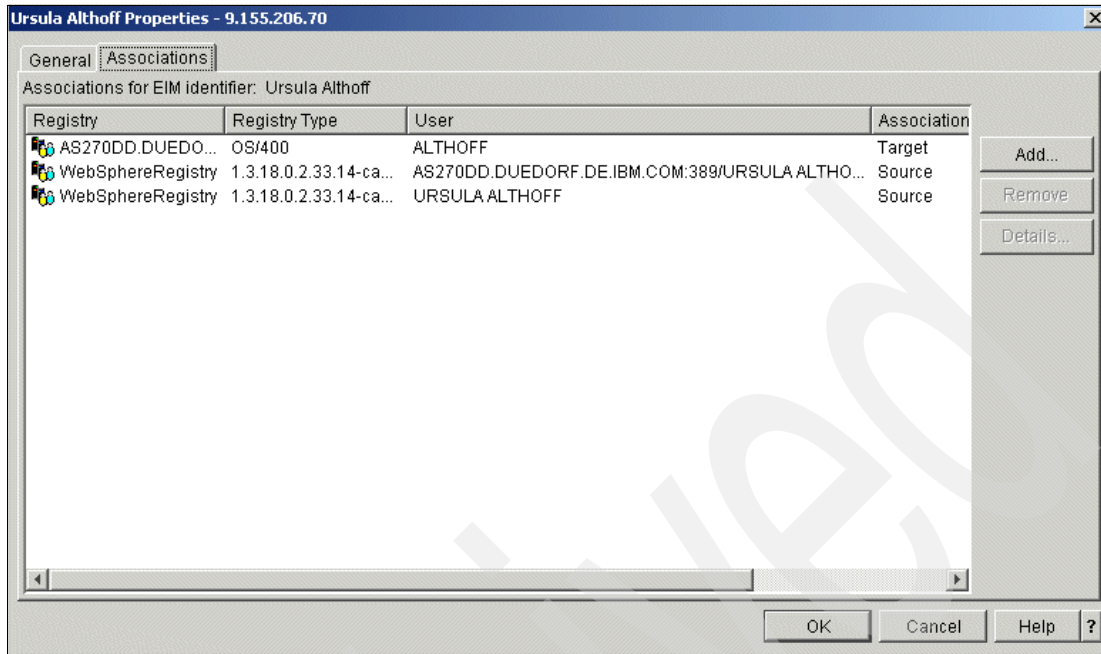


Figure 8-7 EIM Identifier definition for Ursula Althoff

Note: After clicking **Submit**, the request is sent to the IdentityTokenTest servlet, which is protected by a role named allUsers. The allUsers role is bound to the special subject AllAuthenticated, so any user in the WebSphere user registry is authorized to the IdentityTokenTest servlet.


6. Click **OK**.
7. If you specified test1 in the CRTLIB command, then the response should be similar to that as shown in Figure 8-8.

Identity Token Test Results

Library TEST1 created.

Figure 8-8 EIM Identity Token sample application - Result

8. Verify that the library was created under the user profile mapped by EIM:
 - a. From an OS/400 command line type `wrk1nk '/QSYS.LIB/test1.lib'` and press Enter.
 - b. In the Work with Object Links window, type 8 (Display attributes) in the option field to the left of test1.lib and press Enter.
 - c. Check that the value of the owner attribute for library test1 is the user profile mapped by EIM, in our scenario, ALTHOFF.



Deploy the Order Entry and Customer Inquiry application

In this chapter we describe how to deploy the Order Entry application into the WebSphere Application Server instance. The steps to install the Customer Inquiry application are similar.

9.1 Deploy the Order Entry application

1. From the WebSphere administrative console, select **Applications** → **Install New Application**.
2. Select **Local file system** and specify the path where your EAR file resides, which you exported from the WDSC development environment before (see 7.7, “Export the applications from WDSC” on page 168). Click **Next**.

Note: When you export the EAR file to the iSeries IFS, also select **Local file** in this window, with the letter of your mapped network drive.

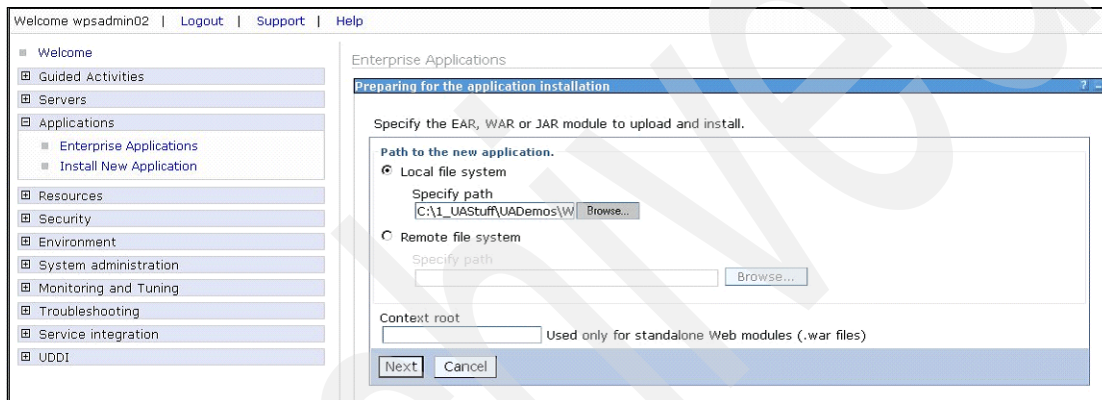


Figure 9-1 Preparing application installation

3. Select the **Use default virtual host name for Web modules** radio button under the Virtual Host section and leave the Host name default_host unchanged. Click **Next**. See Figure 9-2.

Preparing for the application installation

Choose to generate default bindings and mappings.

☐ Generate Default Bindings

Override:

☒ Do not override existing bindings

☐ Override existing bindings

Virtual Host

☐ Do not use default virtual host name for Web modules

☒ Use default virtual host name for Web modules:

Host name
default_host

Specific bindings file
 Browse...

Previous Next Cancel

Figure 9-2 Generate default bindings and mappings

4. In Step 1 of the install wizard, you can change the Application name, as we do, to wfAdvSSO (in our case). Leave all the other properties unchanged and click **Next**. See Figure 9-3.

Install New Application

Specify options for installing enterprise applications and modules.

→ **Step 1: Select installation options**

[Step 2: Map modules to servers](#)

[Step 3: Map resource references to resources](#)

[Step 4: Map virtual hosts for Web modules](#)

[Step 5: Map security roles to users/groups](#)

[Step 6: Summary](#)

Select installation options

Specify the various options that are available to prepare and install your application.

☐ Pre-compile JSP

Directory to install application

☒ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

Application name

☒ Create MBeans for resources

☐ Enable class reloading

Reload interval in seconds

☐ Deploy Web services

Validate Input off/warn/fail

☒ Process embedded configuration

Figure 9-3 Step 1 Select installation options

5. In Step 2, map your application module to the WebSphere Application Server profile and the HTTP server. Select the check box beside your application module (wfAdvSSO) and select both servers that are shown under Cluster and Servers. Click **Apply**. Be sure that both servers are shown in the Server column of the wfAdvSSO module row in the table, as shown in the Figure 9-4. Click **Next**.

Specify options for installing enterprise applications and modules.

Step 2: Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that will serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated based on the applications which are routed through it.

Clusters and Servers:

WebSphere:cell=AS270DD_WFSSOV6,node=IHS_AS270DD_WFSSOV6_node,server=IHS_AS270DD_WFSSOV6
WebSphere:cell=AS270DD_WFSSOV6,node=AS270DD_WFSSOV6_node,server=WFSSOV6

Select	Module	URI	Server
<input checked="" type="checkbox"/>	wfAdvSSO	wfAdvSSO.war,WEB-INF/web.xml	WebSphere:cell=AS270DD_WFSSOV6,node=IHS_AS270DD_WFSSOV6_node,server=IHS_AS270DD_WFSSOV6 WebSphere:cell=AS270DD_WFSSOV6,node=AS270DD_WFSSOV6_node,server=WFSSOV6
<input checked="" type="checkbox"/>	Identity Token Connector	eimIdTokenRA.rar,META-INF/ra.xml	WebSphere:cell=AS270DD_WFSSOV6,node=AS270DD_WFSSOV6_node,server=WFSSOV6

Previous Next Cancel

Figure 9-4 Step 2 Map modules to servers

6. In Step 3, map the resource reference to your application module and the authentication method.
 - a. Select the check box beside your application module in the table at the end of the window and then select the JNDI name eis/idTokenRoot, which is the name you defined for the CF3 connection factory (see 6.4.2, “Configuring an additional connection factory” on page 138). Click **Apply**.

Specify options for installing enterprise applications and modules.

Step 3: Map resource references to resources

Each resource reference that is defined in your application must be mapped to a resource.

javax.resource.cci.ConnectionFactory

To set multiple existing resource JNDI names:

1. Select one or more checkboxes in the table
2. Select existing resource JNDI name
3. Click Apply

Specify existing Resource JNDI name:

eis/idTokenRoot Apply

Figure 9-5 Step 3 Set the resource JNDI name

- b. Second, select the check box beside your application module in the table at the end of the window again. In the Authentication Method section of the window, check the **Use default method** check box and select the idTokenAlias that you defined in 5.2.2, “J2C Authentication Data Entries” on page 97.

Be sure the value for the JNDI name is eis/idTokenRoot and the authentication method is Default Principal Mapping, idTokenAlias before you click **Next**.

Specify authentication method:

☐ none
☒ Use default method
 Select authentication data entry
 idTokenAlias
☐ Use custom login configuration
 Select application login configuration
 Select...

Apply

Select	Module	EJB	URI	Reference binding	JNDI name	Login configuration
<input checked="" type="checkbox"/>	wfAdvSSO		wfAdvSSO.war, WEB-INF/web.xml	eim_wf	eis/idTokenRoot	Resource authorization: Container Authentication method: DefaultPrincipalMapping idTokenAlias

Figure 9-6 Step 3 Specify Authentication Methods

7. In Step 4, Map virtual Hosts for Web modules, accept the default setting and click **Next**. See Figure 9-7.

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

Step 3 Map resource references to resources

→ Step 4: Map virtual hosts for Web modules

Step 5 Map security roles to users/groups

Step 6 Summary

Map virtual hosts for Web modules

Specify the virtual host where you want to install the Web modules contained in your application. You can install Web modules on the same virtual host or disperse them among several hosts.

☒ Apply Multiple Mappings

Select	Web module	Virtual host
<input type="checkbox"/>	wfAdvSSO	default_host ▾

Previous

Next

Cancel

Figure 9-7 Step 4 Map virtual hosts for Web modules

8. In Step 5. Map security roles to users/groups. accept the settings and click **Next**. Because the security role is already defined in the application deployment descriptor, no change is necessary here. Remember you defined the security role and security constraint in 7.2, “Setting up security roles and constraints for your application” on page 151. See Figure 9-8.

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

Step 3 Map resource references to resources



Step 4 Map virtual hosts for Web modules

→ Step 5: Map security roles to users/groups

Step 6 Summary

Map security roles to users/groups

Each role that is defined in the application or module must map to a user or group from the domain user registry.

Select	Role	Everyone?	All authenticated?	Mapped users	Mapped groups
<input type="checkbox"/>	All application users	<input type="checkbox"/>	<input checked="" type="checkbox"/>		

Figure 9-8 Step 5 Map security roles to users/groups

9. Step 6 shows the summary of your deployment definitions (Figure 9-9 on page 187). Click **Finish**.

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

Step 3 Map resource references to resources

Step 4 Map virtual hosts for Web modules

Step 5 Map security roles to users/groups

→ Step 6: Summary

Summary

Summary of installation options

Options	Values
Use Binary Configuration	No
Create MBeans for resources	Yes
Cell/Node/Server	Click here
Reload interval in seconds	
Enable class reloading	No
Process embedded configuration	Yes
Application name	wfAdvSSO1
Validate Input off/warn/fail	warn
Application Scoped Resources	Yes
Directory to install application	
Distribute application	Yes
Deploy Web services	No
Pre-compile JSP	No
Deploy enterprise beans	No

Previous

Finish

Cancel

Figure 9-9 Deployment Summary

10. Once the deployment of the Order Entry (WebFacing) application starts, you will see messages scroll down the window. When the application is installed successfully, click the link **Save to Master Configuration** at the bottom of the page.

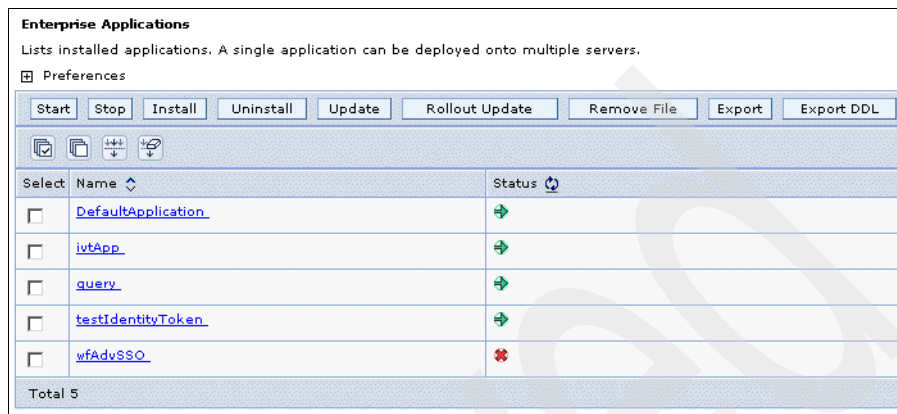
Deploy the Customer Inquiry application in the same way, but in Step 3 of the application installation wizard, use the JNDI name eis/IdentityToken, which is the name you defined for the CF1 connection factory (see 5.2.4, “Connection factories” on page 102).

9.2 Start the new deployed application

After the installation of a new application into the WebSphere Application Server profile, start this new application in the following way:

1. From the WebSphere administrative console, select **Applications** → **Enterprise Applications**.

- You will see your new application (wfAdvSSO) listed in the table (see Figure 9-10). The red cross sign in the Status column of the table indicates that the application is stopped.



Enterprise Applications
Lists installed applications. A single application can be deployed onto multiple servers.

Preferences

Start Stop Install Uninstall Update Rollout Update Remove File Export Export DDL

Select	Name	Status
<input type="checkbox"/>	DefaultApplication	
<input type="checkbox"/>	ivtApp	
<input type="checkbox"/>	query	
<input type="checkbox"/>	testIdentityToken	
<input type="checkbox"/>	wfAdvSSO	

Total 5

Figure 9-10 Enterprise Applications

- Select the check box beside your application and click **Start**.
You should now see the green arrow in the Status in column of the table indicating that the application has started.



Single sign-on sample scenario

In this appendix, we give a short overview on how the workflow of our single sign-on sample scenario looks.

Overview of our scenario

The Customer Inquiry and the Order Entry applications are configured along with the WebSphere Application Server, LDAP, and EIM so that the user, John Day, only has to log on to the system once while using both applications.

Steps 1 through 6 refer to the steps in Figure A-1 on page 191.

1. John Day logs in to the Customer Inquiry (CI) Web tools application in the browser with the credentials John Day.
2. A request is sent through the internet and an HTTP server to the IBM WebSphere Application Server, which in this case is running on the System i5 server.
3. WebSphere Application Server authenticates John (John Day) using the LDAP registry, which stores the users' account details, such as user ID and password.
4. The CI application requests an ID token for the person who was authenticated with the WebSphere Application Server in step 3.
5. The credentials obtained in step 4 enable John to access the System i5 system through the CI application.
6. As John works through the CI application, data is sent back to his browser session through the WebSphere Application Server, the HTTP server, and finally the internet.

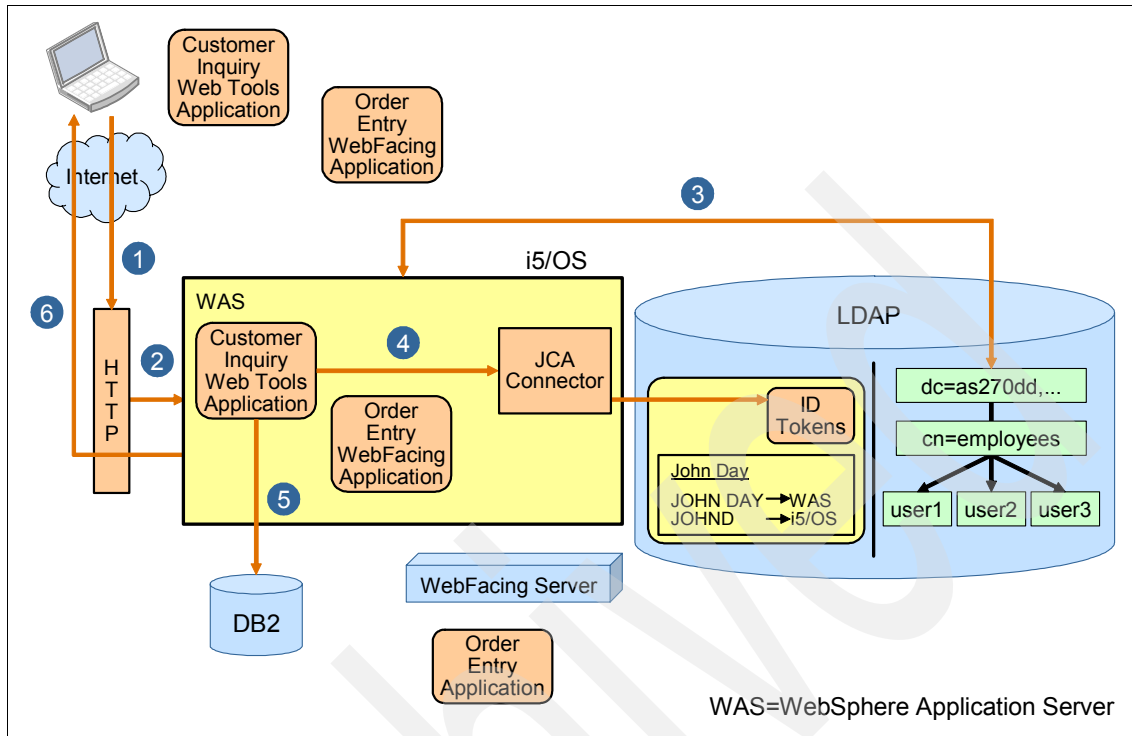


Figure A-1 Customer Inquiry Application Flow

Steps 7 through 9 refer to steps in the Figure A-2.

7. From the CI, John calls the Order Entry (OE) application, and this request ultimately arrives at the WebFacing runtime on the System i5 server.
8. Now, the WebFacing runtime takes the previously authenticated credentials from WebSphere Application Server and makes a request for an ID token for the user that was authenticated to WebSphere Application Server (John Day in this case). This ID token is then passed to EIM.
9. Using this token, EIM “finds” a local i5/OS user ID that represents John, or John Day, on the target System i5 server. EIM returns the ID JOHND to the i5/OS system. Processing then proceeds as though John had provided his i5/OS user profile name and password to the OE application.

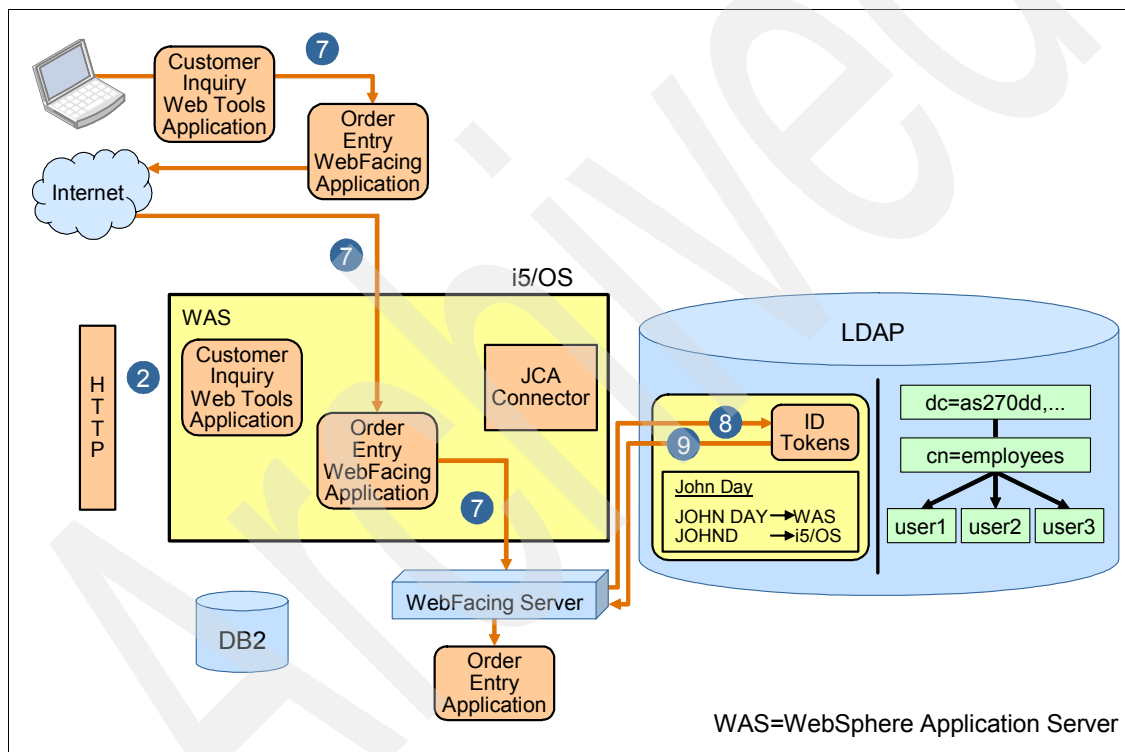


Figure A-2 Customer Inquiry Application Flow 2

Figure A-3 shows the important parameters we have defined in the WebSphere Application Server Definitions that have to map the definitions in the EIM Configuration.

1.

WebSphere Application Server Definitions

Configure Identity Token SSO for Web to i5/OS Access

Identity Token SSO is a mechanism where a single user signon action permits access to multiple i5/OS servers. This allows your Web-based interfaces to access i5/OS back-end applications without having to prompt for additional authentication. Identity Tokens are implemented using Enterprise Identity Mapping (EIM). EIM maintains the relationships between Web users and i5/OS user profiles. The application server creates a token for the servers configured to support Identity Tokens in this EIM Domain.

Note: EIM is hosted on an LDAP server that must be configured and running before continuing.

Configure Identity Tokens:

☐ Do not configure Identity Tokens

☒ Configure Identity Tokens

In order to configure Identity Tokens, an EIM Domain must be configured and the EIM Domain Controller must be running. Specify the LDAP server that is hosting EIM:

LDAP server host name: AS270DD.DUEDORF.DE.IBM.COM e.g. "hostname.domain.com"

LDAP port: 389

LDAP administrator DN: cn=adminstrator e.g. cn=adminstrator

LDAP administrator password: *****

Note: For more information on configuring EIM and Identity Tokens SSO and the required PTFs, see the [iSeries Information Center](#).

Back Next Cancel

2.

Create WebSphere Application Server, V6.0

Configure Identity Token EIM Domain Information

The specified LDAP server has been configured as an EIM Domain Controller. Select the correct EIM domain and source registry name that contains the repository of identity information. Identity Tokens will be controlled based on the mappings that have been defined in this location.

Note: In order for the Identity Tokens support to be activated this WebSphere Application Server instance must have global security enabled. Security will need to be manually configured. See the [iSeries Information Center](#) for details.

EIM Domain Name: eim_fts

Source Registry Name: WebSphereRegistry

Back Next Cancel

EIM Definitions

EIM Domain Controller

DOMAIN = EIM_FFTS

Identifier: John Day

User	Association Type	Registry	Registry Type
John Day	Source	WebSphereRegistry	1.3.18.0.2.33.14-caseIgnore
JOHNDD	Target	AS270DD.DUEDORF.DE.IBM.COM	i5/OS

LDAP Host TCP/IP Name
=
AS270DD.DUEDORF.DE.IBM.COM

Figure A-3 Mapping parameters in WebSphere Application Server and EIM

Appendix A. Single sign-on sample scenario 193

Related publications

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

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 195. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Windows-based Single Signon and the EIM Framework on the IBM eServer iSeries Server*, SG24-6975

Online resources

These Web sites are also relevant as further information sources:

- ▶ IBM Systems Magazine, “Developing WebSphere Applications Using iSeries SSO Techniques”, January 2005, found at:
<http://www.ibmssystemsmag.com/i5/january05/developer/8595p1.aspx?ht=developing%20websphere%20applications%20using%20iseries%20sso%20techniques%20developing%20websphere%20applications%20using%20iseries%20sso%20techniques>
- ▶ IBM Systems Magazine, “What the WAS Application Client Can Do For You”, January 2005, found at:
<http://www.ibmssystemsmag.com/i5/january05/internet/9032p1.aspx?ht=what%20the%20was%20application%20client%20can%20do%20for%20you%20what%20the%20was%20application%20client%20can%20do%20for%20you>

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



Redpaper

Enabling WebSphere Application Server with Single Sign-on

Configure EIM

Create a SSO enabled Application Server

Prepare and deploy applications

There was a time when user names and passwords offered an elegant solution to security concerns. However, this was also the time when companies were just beginning to merge their core processes with technology and had few computer-based applications.

Fast forward to today, when the number of computer applications used daily has surged, and suddenly its elegance has disappeared. Additionally, the popularity of IT solutions (for example, working remotely and Web-based business applications) has increased the demand for secure systems. Consequently, the number of passwords has skyrocketed.

This IBM Redpaper will walk you through installing and configuring an application on a WebSphere Application Server that takes advantage of single sign-on. We begin with an overview of Enterprise Identity Mapping (EIM) and how to set it up. Then we set up the application server and enable the applications. The final steps lead you through the process to deploy and use the sample applications provided with the WebSphere Application Server.

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