Web Express Logon

IBM WebSphere Host On-Demand

What is the purpose of this document?

This document for Host On-Demand administrators who want to do one or more of the following:

- understand how Web Express Logon works
- learn the benefits of implementing Web Express Logon
- learn how Web Express Logon is different than Express Logon Feature (ELF)
- find out if Web Express Logon is appropriate for their environment
- follow step-by-step scenarios of implementing Web Express Logon

We start by explaining the two types of Web Express Logon (macro-based automation and connection-based automation) and how to determine which one is more appropriate for your company's environment. If you decide that your environment does not meet the requirements of either of these two types, you can use the Customizing Web Express Logon section on page 124 to customize your own version of Web Express Logon.

You will see the following two icons throughout this document:

This icon indicates important points that supplement the main text.

This icon refers to information that is specific to FTP sessions.

Table of Contents

What is Web Express Logon?

Web Express Logon is a new feature of Host On-Demand V8 that provides an automated way for users to log on to hosts and host-based applications without having to provide an additional user ID and password. It is designed to function within a wide range of computing environments. Your particular environment, including your host platform and your existing process for maintaining security, determines the way in which you plan for, implement, and use Web Express Logon.

What are the benefits of Web Express Logon?

Benefits of Web Express Logon include the following:

- **Ease of use**: Users can log on to their network security application and access host applications without having to re-enter their IDs and passwords.
- **Reduced password-related support calls**: Users are less likely to call the company support line because of forgotten or misplaced passwords.
- **Increased productivity**: Users can log on only once in an environment that has multiple methodologies for defining user IDs, passwords, and authentications.

How is Web Express Logon different than Express Logon Feature?

Express Logon Feature (ELF) has been available since Host On-Demand V5 and is still available today. However, to better differentiate it from Web Express Logon, we now refer to ELF as Certificate Express Logon. Certificate Express Logon functions the same as ELF did in earlier versions and requires the same configuration. Currently, Web Express Logon and Certificate Express Logon are the two types of Express Logon available with Host On-Demand V8.

Although both Web Express Logon and Certificate Express Logon allow users to log on to host systems without having to enter their user IDs and passwords, the two types of Express Logon have different requirements. For example, Certificate Express Logon requires client-side certificates for user authentication and works exclusively with 3270 session types. In order to use Certificate Express Logon, the client must have a valid client certificate, and the SSL connection must be made to one of the supported TN3270 servers. Web Express Logon, however, does not require client-side certificates, and it can function with most Host On-Demand session types. Which type of Express Logon you choose depends on your environment and your company needs.

For more information about Certificate Express Logon, refer to the Setting up and Using the IBM Express Logon Feature white paper on the Host On-Demand library page at http://www.ibm.com/software/webservers/hostondemand/library.html.

Although Host On-Demand has renamed Express Logon Feature (ELF) to Certificate Express Logon, current documentation outside of Host On-Demand still refers to it as ELF.

How does Web Express Logon work?

Web Express Logon currently offers two styles of logon automation:

- macro-based automation
- connection-based automation

The style of logon automation that best suits your environment depends on your host and session type. If your host allows the client to supply the needed host credentials at the time the connection is established (for example, during the telnet negotiation via a Kerberos passticket), connection-based automation is the appropriate style to use. However, if the client does not receive the needed credentials at time the connection is established, the host must send a login

screen to authenticate the client. Since automating this login screen requires a macro, macro-based automation is the appropriate style. The macro populates the screen's credential fields with the appropriate user information and then transmits this information to the host for authentication.

Read ahead to learn more about the two types of logon automation.

Macro-based automation: an overview

Macro-based automation is for environments of varying host types that (1) are not using Kerberos for network authentication and (2) already have a network security application in place. As the name implies, it requires you to create a macro to perform logon automation.

Host On-Demand provides out-of-the-box support for the following three network security applications without requiring additional coding:

- IBM Tivoli Access Manager
- Netegrity Siteminder
- Microsoft Active Directory (Windows Domain)

If you have a different network security application, you will need to create your own plug-in to work in your environment. For more information, refer to Customizing Web Express Logon on page 124.

Macro-based automation relies on the following four key components and the interactions that take place among them:

- Credential Mapper Servlet (CMS)
- login macro
- Network Security plug-in
- Host Credential Mapper (HCM) database

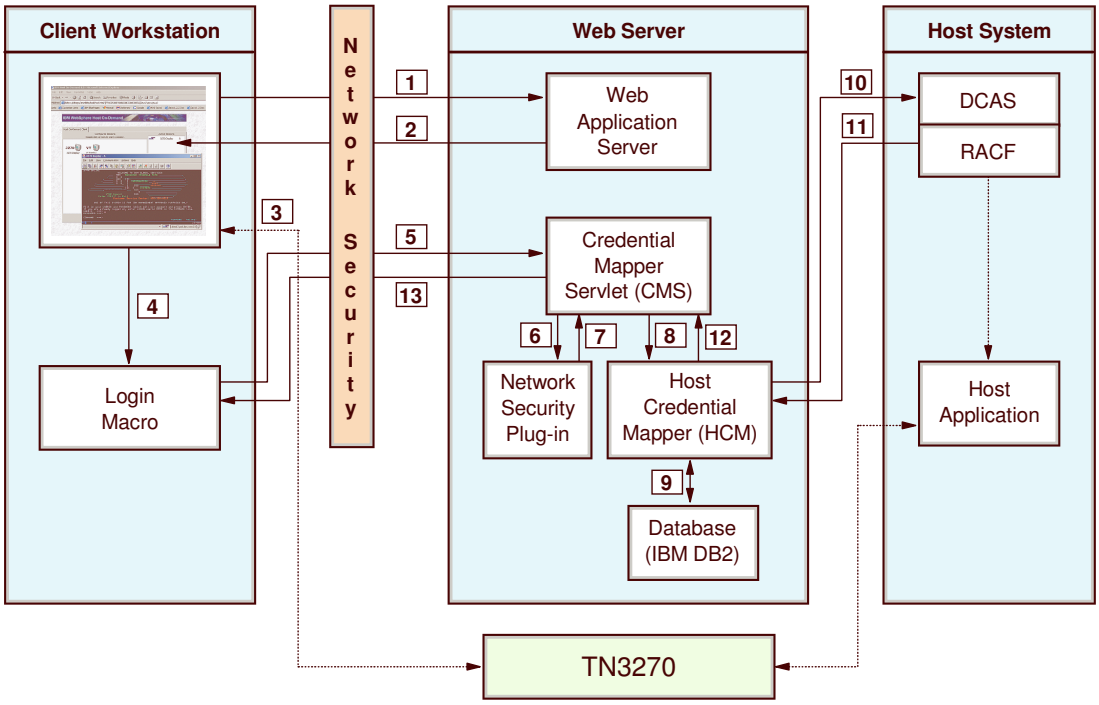
The CMS is supplied with Host On-Demand and must be deployed to a J2EE-compliant Web application server. At a high level, the CMS is responsible for the following tasks: (1) determine the client's identity (called a network ID), (2) map the user's network ID to the host ID, and (3) return the host credentials to the client as an XML document.

The login macro automates the end-to-end process of the client sending the HTTPS request to the CMS, the CMS responding with the needed credentials, and the macro inserting the user's credentials in the proper fields to allow authenticated logon. You must record the login macro while you are in an active session. It initiates at the time the user attempts to access the host session, either automatically or manually (depending on your configuration).

Host On-Demand provides two Network Security plug-ins, one for Tivoli Access Manager and one for Netegrity Siteminder. The Network Security plug-in does not apply to Microsoft Active Directory since the Windows login ID is used as the network ID. The primary function of the Network Security plug-in is to acquire the user's network ID, which may be gleaned from the HTTP header of the incoming HTTP request object.

The HCM database is a back-end repository that maps users' network IDs to their host IDs. This repository can be a JDBC database such as one created with IBM DB2. The Digital Certificate Access Server (DCAS) and Vault plug-ins provided with Web Express Logon are designed to work with such a database. Another possibility for a repository is an LDAP directory. However, using LDAP as your HCM database requires you to write your own plug-in. For more information, refer to Customizing Web Express Logon on page 124.

The following graphic shows you the key components discussed above and how they interact together to achieve logon automation. It illustrates the overall flow of macro-based automation beginning at the point when a user attempts to open a Host On-Demand session and initiates the login macro. If the macro is not configured to auto-start, the user will need to start it manually.



1. The end user clicks a link to the Host On-Demand desktop, which sends an HTTPS request through the network security application to the Web application server.
2. The Web application server returns the HTTPS request and the Host On-Demand desktop displays.
3. The user launches a host session.
4. The login macro executes.
5. The macro sends an HTTPS request to the CMS to obtain the host credentials.
6. The CMS requests the user's network ID from the Network Security plug-in.
7. The Network Security plug-in responds to the CMS with the user's network ID.
8. The CMS passes the network ID and application ID to the HCM plug-in.
9. Using the network and application ID, the HCM plug-in calls upon a database, such as IBM DB2, to map the user's host ID.
10. The HCM plug-in passes the user's host ID and application ID to the host and requests a password or passticket, depending on the type of HCM database. (In this example, The CMS sends the request to DCAS, a TCP/IP server application that interfaces with RACF, a Security Access Facility (SAF)-compliant server product.
11. The host (RACF) identifies the client, checks the client's authorization, and returns the passticket to the HCM plug-in.
12. The HCM plug-in returns the host ID and passticket to the CMS.
13. The CMS returns the host credentials to the client as an XML document.

The login macro automatically inserts the user's credentials in the logon screen fields without user intervention. Now the user is fully authenticated and can proceed with the session.

Connection-based automation: an overview

Unlike macro-based automation, connection-based automation does not require a macro because the client and the host are able to connect without having to provide the user with a login screen. In macro-based automation, a macro is required to automate this screen. Connection-based automation supports the following two environments:

- Telnet-negotiated login
- FTP login

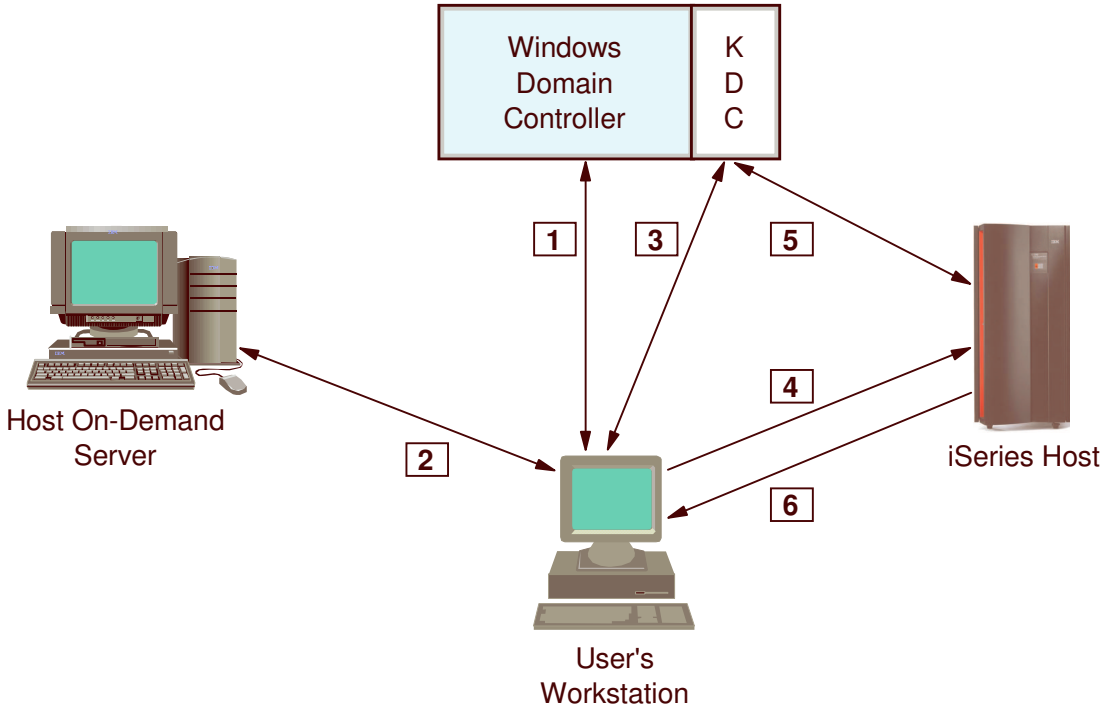
Telnet-negotiated login
Currently, Web Express Logon supports OS/400 (V5R2 and later) telnet-negotiated environments that have Kerberos authentication enabled. It does not require the CMS, a login macro, a Network Security plug-in, nor the HCM database. Instead, it extends the existing single sign-on capability of the OS/400 operating system.

In order for connection-based automation to function in this environment, you must have the following prerequisites in place:

- Windows Domain Controller (Microsoft Active Directory)
- key distribution center (KDC)
- Kerberos network authentication enabled on each target OS/400 system
- OS/400 V5R2 (5722-SS1) or later as the host operating system
- one or more of the following client operating systems:
    - Windows 2000 Professional and Server
    - Windows XP Professional
    - Windows Server 2003

You must configure your OS/400 environment to use single sign-on capability in order to implement connection-based logon automation. The OS/400 environment provides single sign-on capability through a combination of network authentication service (NAS) and an IBM technology called Enterprise Identity Mapping (EIM). Host On-Demand uses this existing methodology for acquiring credentials to allow users to bypass the 5250 session login screen. Both NAS and EIM technology are available with the OS/400 (V5R2 and later) operating system.

The following graphic illustrates the overall process of connection-based automation in an OS/400 environment with Kerberos authentication enabled:



1. A user logs on to the Windows domain. The Windows domain gives users access to the network.
2. The user requests a Host On-Demand session from the Host On-Demand server.
3. The Host On-Demand session initializes and requests a Kerberos ticket from the KDC. This is how users gain access to the individual resources within the network.
4. The user attempts to create a connection with the identified session using the Kerberos ticket as the credential.
5. The iSeries host validates the ticket with the KDC.
6. The user is successfully logged in.

FTP login

Web Express Logon provides an automated way for users to log on to FTP hosts by providing a central repository for storing and retrieving user's credentials. Although this process is similar to configuring Web Express Logon in a vault-style environment (see Scenario 2: Configuring Web Express Logon in a vault-style environment on page 64), this type of automation is different because the user's credentials are retrieved from the CMS at the time the connection is established. In other words, it does not require a macro. Currently, Host On-Demand allows you to statically store a user's ID and password in the FTP configuration; however, Web Express Logon extends this approach by automating the user credential retrieval process.

In order to enable Web Express Logon for FTP sessions, follow the steps for Scenario #2: Configuring Web Express Logon in a vault-style environment on page 64. Look for the following icon to provide you specific information that applies to FTP sessions.

Overview of three real-life implementation scenarios

The following three scenarios are based on real-life examples and are designed to help you successfully implement Web Express Logon. The first two scenarios are for macro-based automation and the third one is for connection-based automation.

Scenario #1: Macro-based automation: Configuring Web Express Logon in a z/OS and DCAS environment

In this scenario, you are an administrator who works for an electric and gas company with over 1000 employees. These employees must connect to z/OS V1R4 host systems and host-based applications throughout the day to access customer and employee records. Your job is to maintain this environment.

In addition to configuring your network security application, you must edit and deploy the CMS provided with Host On-Demand, create your HCM database, and configure DCAS to work with Web Express Logon. Once that is complete, you use the Deployment Wizard to create your HTML file, configure your 3270 host session, and record your login macro.

Here is a summary of your environment:

- host operating system: z/OS V1R4 with APAR PQ74457
- network security application: IBM Tivoli Access Manager for e-business V4.1
- J2EE-compliant Web application server for editing and deploying CMS: IBM WebSphere Application Server V5
- HCM database application: IBM DB2 Universal Database V7

Scenario #2: Configuring Web Express Logon in a vault-style environment

In this scenario, you are the administrator for a large financial markets/retail banking company. Using Host On-Demand, the company Web-enabled a commercial credit application to provide Internet access to one of the bank's largest customers, a large automobile retailer. This Web-enablement provides better and faster service to the commercial customer who, in turn, provides better and faster service to their customers.

You have just upgraded to Host On-Demand V8 and are planning to implement Web Express Logon so your customers who work at the automobile retailer do not have to log on manually to the commercial credit application, which resides on a z/OS V1R3 host system. In order for this to happen, you must configure your network security application, edit and deploy the CMS provided with Host On-Demand, and create your HCM database. Once that is complete, you use the Deployment Wizard to create your HTML file, configure your 3270 host session, and record your login macro.

Here is a summary of your environment:

- host operating system: z/OS V1R3
- network security application: IBM Tivoli Access Manager for e-business V4.1
- J2EE-compliant Web application server for editing and deploying CMS: IBM WebSphere Application Server V5
- HCM database application: IBM DB2 Universal Database V7

Within this scenario, you will see this FTP icon to highlight points that relate specifically to enabling Web Express Logon for FTP sessions.

Scenario #3: Connection-based automation: Configuring Web Express Logon in an OS/400 and Kerberos environment

In this scenario, you are an administrator who manages the network for the shipping and receiving department for a large apparel manufacturer. Throughout the day, hundreds of manufacturer's representatives log on to the network and connect to two iSeries host systems (both running OS/400 V5R2) to access order entries, order status, and shipping and inventory information.

In your company's environment, network users are Windows 2000 clients who access the network through a Windows domain (Microsoft Active Directory). When they attempt to access individual resources (such as a host-based application) on the network, they request access from the key distribution center (KDC), which is a Windows 2000 server that houses a database of user IDs and passwords.

In order to implement Web Express Logon, you must first enable single sign-on capability in your OS/400 environment by implementing network authentication service (NAS), or Kerberos authentication, and then configuring a mapping architecture called Enterprise Identity Mapping (EIM). Together, they create a single sign-on capability. Host On-Demand simply extends this capability.

Scenario #1: Macro-based automation: Configuring Web Express Logon in a z/OS and DCAS environment

You are a network administrator who works for an electric and gas company with over 1000 employees. These employees must connect to z/OS V1R4 host systems and host-based applications throughout the day to access customer and employee records. Using IBM WebSphere Host On-Demand, they are able to access the data securely through their Java-enabled browsers and do not have to interact directly with the traditional mainframe green screen. Your job is to maintain this environment.

Here is a summary of your environment:

- host operating system: z/OS V1R4 with APAR PQ74457
- network security application: IBM Tivoli Access Manager for e-business V4.1
- J2EE-compliant Web application server for editing and deploying CMS: IBM WebSphere Application Server V5
- HCM database application: IBM DB2 Universal Database V7

You have just upgraded to Host On-Demand Version 8 and learned that you can enable single sign-on for Host On-Demand within your existing z/OS V1R4 environment.

You take the following steps to enable Web Express Logon:

1. Complete the planning worksheet.
2. Configure the network security application (if needed).
3. Configure the Digital Certificate Access Server.
4. Create the SSL key database.
5. Create the Host Credential Mapper database.
6. Configure the Credential Mapper Servlet.
7. Deploy the Credential Mapper Servlet.
8. Start creating your HTML file.
9. Configure your Host On-Demand session.
10. Record the Web Express Logon macro.
11. Finish creating your HTML file.

Step 1 of 11: Complete the planning worksheet

The following questions illustrate the type of information you will need before you begin configuring Web Express Logon. The reponses that the administrator gave in this scenario are in the Answers column.

Questions	Answers
Is your host type z/OS V1R4 with APAR PQ74457 (required for DCAS)?	Yes
Which network security application do your users go through to access the network?	IBM Tivoli Access Manager for e-business V4.1
Which J2EE-compliant Web application server will you use to edit and deploy your CMS?	IBM WebSphere Application Server V5
Which application will you use to store HCM values?	IBM DB2 Universal Database V7

Step 2 of 11: Configure the network security application (if needed).

If you are using one of the three network security applications that Web Express Logon supports — IBM Tivoli Access Manager, Netegrity Siteminder, or Microsoft Active Directory — you may not need any additional configuration. This document assumes that you already have a network security application in place and have configured any additional steps needed to allow Web Express Logon's Network Security plug-in to acquire the user's network ID. Recall that once Host On-Demand acquires the user's network ID, the HCM database maps it to the user's host ID in order to acheive logon automation. If the plug-in cannot acquire this network ID, single sign-on capability will be lost.

In this scenario, the administator has already installed IBM Tivoli Access Manager but needs to perform some additional configuration in order for the Network Security plug-in to acquire the user's network ID successfully. This additional configuration involves WebSEAL, the resource manager component of Tivoli Access Manager that is responsible for inserting the user's network ID into the HTTP header as it passes the request on to the destination host.

In order for WebSEAL to insert the user's network ID into the HTTP header, the administrator must create a junction with a `-c all` option included. To create the junction, he logs in as the sec_master administration user and issues the following `pdadmin> server` task command:

```
pdadmin> server task webseald-cruz create -f -c all -w -t tcp -h
dtawg.raleigh.ibm.com -p 80 /junction
```

where *`webseald-cruz`* is the name of the Tivoli Access Manager server host name, *`dtawg.raleigh.ibm.com`* is the fully qualified domain name of the back-end server, *`80`* is the port number (this is the default), and *`junction`* is the name of the junction point.

For more information about Tivoli Access Manager, WebSEAL, and creating WebSEAL junctions, refer to the following Web site:

http://publib.boulder.ibm.com/tividd/td/IBMAccessManagerfore-business4.1.html

Step 3 of 11: Configure the Digital Certificate Access Server.

The Digital Certificate Access Server (DCAS) is a TCP/IP server application that runs on OS/390 V2R10 and later (z/OS included). It interfaces with a Security Access Facility (SAF)-compliant server product to assist with express logon services such as Web Express Logon. In this scenario, this SAF-compliant server product is IBM Resource Access Control Facility (RACF).

In this scenario, the administrator must configure the DCAS application and RACF on the z/OS server to work with Web Express Logon. He must also create an SSL key database file that contains both the DCAS client certificate information and the DCAS server's certificate (public key) information. He will create this file in the next step.

For more information, refer to the z/OS V1R4.0 Communications Server IP Configuration Reference at http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/Shelves/EZ2ZO108, publication number SC31-8776-03. Also refer to the z/OS V1R4 APAR PQ74457 for information on how to configure the DCAS server to function with Web Express Logon.

To configure DCAS and RACF, the administrator takes the following steps:

A. Configure RACF services for the DCAS
B. DCAS and system SSL
C. Authenticate the DCAS and the DCAS client
D. Manage keys and certificates using RACF's Common key ring support
E. Define a passticket profile for each application
F. Configure the DCAS
G. Start the DCAS

****

A. Configure RACF services for the DCAS: This section describes how to configure RACF services for the DCAS, including the following three subtopics:

i. Define a User ID as superuser to OMVS Services
ii. Provide a User ID with Access to MVS.SERVMGR.DCAS
iii. Provide a RACF Definition for MVS Startup

In the following example RACF commands, italicized items should be replaced with values appropriate for your environment.

i. *Define a User ID as superuser to OMVS Services*:
The DCAS server runs as a system daemon and must be started under a controlled user ID that has superuser authority (meaning, not an end-user or system programmer user ID). To define the user ID to use OMVS services, use the following command:

```
ADDUSER dcasid DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/'))
```

where *dcasid* is the name of the user ID.

ii. *Provide a User ID with Access to MVS.SERVMGR.DCAS*:
Starting the DCAS from an MVS procedure requires that the user ID from which it is started have access to the MVS.SERVMGR.DCAS resource in the OPERCMDS class. To provide this access, use the following commands:

```
RDEFINE OPERCMDS (MVS.SERVMGR.DCAS) UACC(NONE)
PERMIT MVS.SERVMGR.DCAS CLASS(OPERCMDS) ACCESS(CONTROL) ID(dcasid)
```

where *dcasid* is the name of the user ID.

iii. *Provide a RACF Definition for MVS Start-up*:
If DCAS is started as an MVS procedure, you will need the following RACF definition:

```
RDEFINE STARTED DCAS.* STDATA(USER(dcasid))
SETR RACLIST(STARTED) REFRESH
```

where *dcasid* is the name of the user ID.

At a minimum, you must use RACF to associate the certificate with a valid user ID. You can do this using the RACDCERT ADD command. The user ID could be the one associated with the DCAS itself or it could be any valid user ID. If you want additional checking, you must activate the SERVAUTH class and define an EZA.DCAS.cvtsysname profile with the user ID associated with the client certificate to access the profile.

B. DCAS and system SSL: This section gives an overview of using System SSL with the DCAS.

The DCAS and the DCAS client use SSL to communicate. The SSL protocol begins with a handshake. Then, the DCAS client authenticates the DCAS and vice versa. At this time, the DCAS and the DCAS client also agree on how to encrypt and decrypt the data.

You can specify the cipher level used for encryption and decryption for each connection at the time DCAS is configured, using the V3CIPHER configuration keyword. Alternatively, you can set the cipher level dynamically when DCAS starts, based on the level of cipher installed on the system. To set the cipher level dynamically, do not specify the V3CIPHER keyword.

SSL provides data privacy and integrity as well as client and server authentication based upon public-key certificates. For each SSL connection, SSL uses a public/private key (PKI) mechanism for authenticating each side of the connection and for agreeing on encryption keys. These keys are generated and stored in key databases, known as key rings.

X.509 certificates, containing public keys, are also required. The X.509 certificates can be created or requested and received. In either case, a certificate is then associated with and becomes part of a key ring. You have access to several services for creating and managing key rings and certificates:

The gskkyman tool
This tool is shipped with System SSL and runs out of the OS/390 and z/OS UNIX shell. You can use it to create key rings and certificates that are stored in Hierarchical File System (HFS). Specify key rings created with gskkyman in the DCAS configuration file using the KEYRING keyword.

If you use gskkyman, you must also create a password stash file. The password stash file protects the key ring file because it contains private keys associated with the certificates contained in the key ring. Specify the password stash file in the DCAS configuration file using the STASHFILE keyword. For details on using the gskkyman tool, refer to the z/OS System Secure Sockets Layer Programming Guide and Reference.

The RACDCERT command
You can also use the RACDCERT command in RACF to create, register, store, and administer keys and certificates. If you use RACDCERT, specify the key ring to the DCAS server in the configuration file using the SAFKEYRING keyword. A key ring created this way does not have a password file associated with it. For details on digital certificates, refer to the z/OS SecureWay

Security Server RACF Security Administrator's Guide and z/OS SecureWay Security Server RACF Command Language Reference.

C. Authenticate the DCAS and the DCAS client: The type of security and authentication required will determine the way certificates are created and managed. The DCAS, in conjunction with RACF, supports several levels of authentication.

Authenticating the DCAS: DCAS authentication is always performed by the DCAS client. Authentication requires that the DCAS receive the user's z/OS application ID and user ID.

Authenticating the DCAS client: The DCAS client interacts with the DCAS. Authenticating the DCAS client involves additional levels of control in which the client must have a key database with a certificate. Depending on the control level, the certificate is authenticated by SSL and the DCAS using RACF services. You will create this SSL key database file in the next step.

There are three levels of client authentication from which to choose:

Level 1
With Level 1 authentication, the DCAS uses the client authentication provided by SSL key database file. This file must contain the following certificates:
– The DCAS certificate
– The DCAS client certificate
To configure DCAS for this level of authentication, specify the CLIENTAUTH LOCAL1 keyword and value in the DCAS configuration file. Use the KEYRING or the SAFKEYRING keywords in the DCAS configuration file to specify the key ring used by the DCAS.

Level 2
Level 2 includes Level 1 authentication plus additional verification that the DCAS client certificate has been associated in RACF with a valid user ID. (This user ID must be the user ID that DCAS is running under.) To configure DCAS for this level of authentication, specify the CLIENTAUTH LOCAL2 keyword and value in the DCAS configuration file. Use FTP (with the BINARY send option) to send the DCAS client's DER certificate to an MVS dataset. Use the RACDCERT ADD command to add the certificate to RACF and associate it with a user ID, as shown in the following example:

```
RACDCERT ID(dcasid) ADD('DCAS.HOSTPUB.CERT') TRUST
```

where *dcasid* is the name of the user ID.

Level 3
Level 3 includes Level 2 authentication, and in addition, it verifies that the DCAS client has access to the DCAS. The user ID derived from the certificate using the RACF checks from Level 2 is defined as having access to the SERVAUTH RACF class and the EZA.DCAS.cvtsysname resource in the SERVAUTH class. The following two conditions apply:

- If the SERVAUTH class is not active or the EZA.DCAS.cvtsysname profile is not defined, or both, it is assumed this enhanced level is not requested.
- If the SERVAUTH class is active and the EZA.DCAS.cvtsysname profile is defined (but not for the user associated with the certificate) the requester's connection is terminated:

```
RDEFINE SERVAUTH EZA.DCAS.cvtsysname UACC(NONE)
PERMIT EZA.DCAS.cvtsysname CLASS(SERVAUTH) ACCESS(CONTROL) ID(dcasid)
```

where *dcasid* is the name of the user ID.

To configure DCAS for Level 3 authentication, follow these steps:

i. Specify the CLIENTAUTH LOCAL2 keyword and value in the DCAS configuration file.
ii. Activate the SERVAUTH RACF class.
iii. Define a profile for the EZA.DCAS.cvtsysname resource and associate the profile with the user ID associated with the certificate.

The ID associated with the certificate and the EZA.DCAS.cvtsysname can be any valid user ID.

D. Manage keys and certificates using RACF's Common key ring support.

For information on RACF commands, refer to z/OS SecureWay Security Server RACF Security Administrator's Guide and z/OS SecureWay Security Server RACF Command Language Reference.

Initial Setup

i. Before using RACF to store your key database information, ensure that the digital certificate and digital key ring (DIGTCERT and DIGTRING) classes are active before defining certificates or key rings to RACF:

```
SETROPTS CLASSACT(DIGTCERT  DIGTRING)
```

ii. Be sure to perform a refresh after each update or change:

```
SETROPTS RACLIST (DIGTRING  DIGTCERT) REFRESH
```

iii. Also, ensure that the RACDCERT command is defined as an authorized TSO command in the IKJTSOxx member.

iv. In order to issue the RACDCERT command, you must have access to the FACILITY class IRR.DIGTCERT.function with UPDATE or CONTROL access. If the DCAS is started as an MVS started procedure, you must permit the RACF user ID to IRR.DIGTCERT.LIST. If the DCAS is started from a TSO user ID under the OS/390 UNIX shell, you must also permit that ID, as shown in the following example:

```
RDEFINE FACILITY (IRR.DIGTCERT.function)
           UACC(NONE)
PERMIT  IRR.DIGTCERT.LIST
           CLASS(FACILITY) ID(dcasid)
           ACCESS(control)
```

where *dcasid* is the name of the user ID.

Create a Key ring:
You will need to create a key ring for your DCAS server. For example:

```
RACDCERT ID(dcasid) ADDRING(SERVERKeyring)
```

where *dcasid* is the name of the user ID.

Create and Connect a Certificate:
You can use RACF to create self-signed certificates (see next page).

Request and Connect a Well-known Certificate:
You can alternately request a well-known certificate from a Certificate Authority, such as Verisign, and add it to RACF (see next page).

Creating and Connecting Self-signed Certificates on the Host
Because the clients will not know about the issuer of the self-signed certificate, in most cases you must add the server's self-signed certificate to the client's signer certificates. This process requires the following high-level steps:

i. Generate the DCAS server self-signed certificate on the host.
ii. Transfer the DCAS server's certificate to the DCAS client machine.

Following are detailed steps describing the process. DCAS server self-signed certificates can be created using RACF or GSKKYMAN.

If using RACF, take the following steps:

i. Generate the DCAS server self-signed certificate on the host and transfer to the DCAS client.

(1) Create a self-signed server certificate using RACDCERT gencert:

```
RACDCERT ID(dcasid)
             SUBJECTSDN(CN('DCASCERT')
              OU('TEST')
              C('US'))
              TRUST
              SIZE(512)
              WITHLABEL('DCASCERT')
```

where *dcasid* is the name of the user ID.

(2) Use RACDCERT Connect to connect the certificate to a key ring and make it default. This example assumes a key ring called SERVERKeyring already has been created.

```
RACDCERT ID(dcasid)
             CONNECT(ID(dcasid)
              LABEL('DCASCERT')
              RING(SERVERKeyring)
              USAGE(PERSONAL) DEFAULT)
```

where *dcasid* is the name of the user ID.

(3) Use RACDCERT EXPORT to export the DCAS server self-signed certificate in ".DER" format to an MVS file.

```
RACDCERT ID(dcasid) EXPORT(LABEL('DCASCERT'))
            DSN('dcasid.SAFCERT.DER')
            FORMAT(CERTDER)
```

where *dcasid* is the name of the user ID.

ii. FTP the exported DCAS server certificate to the DCAS client using the FTP binary option.

If using a GSKKYMAN key ring, take the following steps:

i. Generate the DCAS server self-signed certificate on the host.
(1) Open your key ring file and select 'Create a self-signed certificate.'
(2) Specify Version 3, label, key size, and certificate information when requested.

(3) Set the key as the default in your key database.
(4) Save the certificate to a file, select binary format (the certificate will be saved in binary ".DER" format).

The following is a sample of GSKKYMAN output for creating a self-signed certificate. GSKKYMAN's default action appears in the brackets.

(1) Enter version number of the certificate to be created (1,2, or 3) [3]: 3
(2) Enter a label for this key.....> selfsignedcert
(3) Select desired key size from the following options (512): 1: 512 2: 1024
(4) Enter the number corresponding to the key size you want: 1
(5) Enter certificate subject name fields in the following.

Common name (required).....>test server certificate
Organization (required).....>dev
Organization Unit (optional).....>
City/Locality (optional).....>
State/Province (optional).....>
Country Name (required 2 characters).....>US

(6) Enter number of valid days for the certificate[356]:
(7) Do you want to set the key as the default in your key database? (1=yes, 0=no) [1]: 1
(8) Do you want to save the certificate to a file? (1=yes, 0=no) [1]:
(9) Should the certificate binary data or Base64 encoded ASCII data be saved? (1=ASCII, 2=binary) [1]: 2
(10) Enter certificate file name or press ENTER for "cert.crt": ss-servercert.crt
The following message is displayed: Please wait while the self-signed certificate is created.

To pick up the new default server certificates, restart TCP/IP or stop all secure ports and issue a VARY OBEY command to bring the secure ports back online.

ii. Transfer the host's certificate to the DCAS client machine. If using FTP, transfer with the binary ftp option.

Creating and Connecting Well-known Certificates on the Host
Following are the steps for adding a Certificate Authority Root and Personal Certificates to the Host.

i. Create a self-signed certificate and key pair for the DCAS server:

```
RACDCERT ID(dcasid)
GENCERT SUBJECTSDN(CN('labelname') C('us'))
WITHLABEL('labelname')
```

where *dcasid* is the name of the user ID.

ii. Create a certificate request for a Certificate Authority (CA) by issuing RACDCERT GENREQ against the self-signed certificate:

```
RACDCERT ID(dcasid)
   GENREQ(LABEL('labelname'))
   DSN(labelname.certreqname)
```

where *dcasid* is the name of the user ID.

iii. Send the certificate request to a Certificate Authority. (For example, IBM Trust Authority, Entrust, Verisign.)

iv. When you receive the DCAS server certificate from the Certificate Authority, transfer the file to the DCAS host.

v. If RACF doesn't already have the root certificate for the Certificate Authority, then you need to get it in .DER format, and add it to RACF using this command:

```
RACDCERT CERTAUTH ADD(caroot.der)
   TRUST WITHLABEL('caroot')
```

vi. Add the DCAS server certificate from the Certificate Authority back into RACF:

```
RACDCERT ID(dcasid) ADD(certname) WITHLABEL('certname')
```

where *dcasid* is the name of the user ID.

vii. Connect the CA root certificate to the key ring with usage CERTAUTH:

```
RACDCERT ID(dcasid)
   CONNECT(CERTAUTH LABEL('caroot')
   RING(SERVERKeyring)
   USAGE(CERTAUTH) DEFAULT)
```

where *dcasid* is the name of the user ID.

viii. Connect the DCAS server certificate to the key ring with usage PERSONAL:

```
RACDCERT ID(dcasid)
   CONNECT(ID(dcasid) LABEL('certname')
   RING(SERVERKeyring)
   USAGE(PERSONAL) DEFAULT)
```

where *dcasid* is the name of the user ID.

E. Define a passticket profile for each application: You must create a RACF PTKTDATA (passticket data class profile) for each application ID. This profile allows the DCAS to obtain a passticket for the application and user ID and to pass it back to the client. This profile name must match the RACF PTKTDATA application name that is configured on the host. This name could be the same as the application name that the user is logging onto (for example, the name on USSMSG10).

When creating PTKTDATA profiles for applications such as TSO, the application name portion of the profile will most likely not be the same. For example, RACF requires that the application ID portion of the profile name be TSO+SID. Refer to z/OS SecureWay Security Server RACF Security Administrator's Guide to determine the correct profile naming.

You must create these profiles on each separate RACF system (the system where the users will be logging on to) that contains target applications for Web Express Logon.The PTKTDATA class profile defined in the "target" RACF system must match the PTKTDATA class profile in the system where the passticket is created, which in the case of Web Express Logon, is the system where the DCAS server executes. These PTKTDATA class profiles need to have corresponding profile names and identical secret keys (defined using the KEYMASKED parameter).

An example of a passticket data class profile for the application TSORUS (the KEYMASKED value is a hexadecimal string of your choice) is as follows:

```
RDEFINE PTKTDATA TSORUS
SSIGNON(KEYMASKED(E1E2E3E4E5E6E7E8)
UACCESS(NONE) )
SETR RACLIST(PTKTDATA) REFRESH
```

Pay special attention to the APPLID name. For example, for TSO, the profile is TSO+SID. The SID is the SMF system id that is defined in the SMFPRMxx member in SYS1.PARMLIB. For more information on defining passticket profiles, refer to the z/OS SecureWay Security Server RACF Security Administrator's Guide.

F. Configure the DCAS: Make sure the DCAS configuration file and the DCAS start procedure are updated as appropriate to your installation. The DCAS configuration file (/etc/dcas.conf) contains the following keywords:

TCPIP tcpstackname	; Server will have affinity to tcpip stackname
IPADDR xx.xx.xx.xx	; IP address used to bind to for SSL connection (defaults to inaddr_any)
PORT xxxx	; DCAS listens on this port number (default is 8990)
KEYRING /etc/ssl/xxx.kdb	; HFS file name of Keyring for SSL negotiation
STASHFILE /etc/ssl/xx.sth	; Stash file containing the Password of Keyring file
SAFKEYRING SERVERKeyring	; Key ring via RACF
CLIENTAUTH xxxxxx	; Client Authentication level ; LOCAL1 (SSL does authentication) ; LOCAL2 (default - use RACF to validate the client's certificate)
LDAPSERVER xx.xx.xx.xx	; Fully qualified name or IP address of LDAP Server
LDAPPORT xxxx	; Port# that LDAP Server is listening on
V3CIPHER cipherspec	; Specify a subset of the supported SSL V3 cipher algorithms ; The following cipher levels are valid: ; 01=NULL MD5 02= NULL SHA 03=RC4 MD5 Export ; 04=RC4 MD5 US 05=RC4 SHA US 06=RC2 MD5 Export ; 09=DES SHA 0A=Triple DES SHA US
SERVERTYPE	; The SERVERTYPE paramter determines the type of functions that DCAS will support for connecting clients. Valid SERVERTYPE values include the following:

	``; ALLTYPES - Supports any of the available DCAS functions ; CERTTYPE - DCAS will accept an x.509 certificate and application name and provide a userid and passticket. This is the default. ; NOCERTTYPE DCAS will not accept an x.509 certificate and application name (as supported for the Express Logon Feature). May be used to turn off CERTTYPE when a previous SERVERTYPE CERTTYPE or ALLTYPES was specified. *Note*: A specification of SERVERTYPE NOCERTTYPE by itself is not allowed, since it will turn off the default value. ; USERIDTYPE - DCAS will accept a userid and application name and provide a passticket ; NOUSERIDTYPE DCAS will not accept a userid and application name. May be used to turn off USERIDTYPE when a previous SERVER TYPE USERIDTYPE or ALLTYPES was specified.```

G. Start the DCAS: You can start the DCAS as either a generic server without stack affinity or as a server with affinity to a specific TCP/IP stack. You can start the DCAS in the following three ways:

- automatically when the TCP/IP address space is started
- from the z/OS UNIX shell
- from an MVS started procedure

To start the DCAS automatically when the TCP/IP address space is started, specify DCAS on the AUTOLOG statement in the TCPIP profile dataset as shown in the following example:

```
AUTOLOG
DCAS
ENDAUTOLOG
```

Following is a sample procedure used to start DCAS. First, enter the command S DCAS. To pass optional parameters to DCAS, specify them after the final slash (/) on the PARM statement, for example:

```
// PARM=('POSIX(ON) ALL31(ON)'
// 'ENVAR("LIBPATH=/usr/lib")/-d 3 -l SYSLOGD')
```

Sample procedure:

```
//DCAS PROC
//* DEBUGGING AND LOGGING MAY BE REQUIRED TO HELP DETERMINE A PROBLEM
//* THE DCAS.
//*
//* -D OR -D - INDICATES DEBUGGING LEVEL REQUESTED.
//* FORMAT: -D LEVEL
//* LEVEL IS: 1=LOG ERROR AND WARNING MESSAGES
//* 2=LOG ERROR, WARNING, AND INFO
//* 3=LOG ERROR,WARNING, INFORMATI
//*
//*<BR>//DCAS EXECPGM=EZADCDMN,REGION=4096K,TIME=NOLIMIT,
// PARM='POSIX(ON) ALL31(ON) / -d 3 -l SYSLOGD'
```

```
//*
//SYDENV DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSERR DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//*
```

You will find a sample start procedure in EZADCASP in the SEZAINST dataset.

To start the DCAS from the z/OS UNIX shell, use the following format:

```
dcas <parameter_1> <parameter_2> <parameter_3> &
```

To start the DCAS from an MVS started procedure, use the following format:

```
PARM=.../<parameter_1> <parameter_2> <parameter_3>
```

You can use the following optional parameters from both the z/OS UNIX shell and the MVS started procedure:

-d or -D
Indicates debugging. The following levels apply:
1= Specifies log error and warning messages.
2= Specifies log error, warning, and informational messages.
3= Specifies log error, warning, informational, and debug messages. This is the default.

-l or -L
Indicates logging to SYSLOGD or to a designated log file. If you do not specify this parameter, logging defaults to /tmp/dcas.log. If you specify a debug level, but not logging, then the DCAS attempts to open the default log file /tmp/dcas.log. If this fails, debugging is turned off. For SYSLOGD, the DCAS uses the log facility local0.

-c or -C
Indicates the requested configuration file (for example, /u/userx/passtick.conf). If you do not specify this parameter, the DCAS looks for the configuration file using the following search order: DCAS_CONFIG_FILE environment variable /etc/dcas.conf tsouserid.DCAS.CONF TCPIP.DCAS.CONF If the DCAS does not find a valid configuration file, it will not start.

When DCAS is started, it stores its process ID (pid) in an HFS file. The file name under which it is stored depends upon how you configure DCAS:

- If you configure the DCAS with TCP/IP stack affinity, the pid file is named /tmp/dcas.tcpipname.pid, where tcpipname is the name of the TCP/IP stack for which DCAS has affinity.
- If you configure the DCAS without stack affinity, pid file is named /tmp/dcas.INET.pid.

You can stop the DCAS from the UNIX shell or from MVS:

- To stop the DCAS from the UNIX shell, use the following command: `kill -s SIGTERM pid`
- To stop the DCAS from MVS, use the following command: `P DCAS`

Step 4 of 11: Create the SSL key database file.

Now that you have configured the DCAS, you need to provide the CMS the ability to establish a secure connection to the DCAS. This secure connection takes place via a key database file that stores certificates that authenticate the DCAS server and optionally authenticate the CMS.

To allow the CMS to authenticate the DCAS server when it establishes an SSL connection, you must store the DCAS server's certificate (public key) information in the key database file. Optionally, to authenticate the CMS, you must add a client certificate to the key database file.

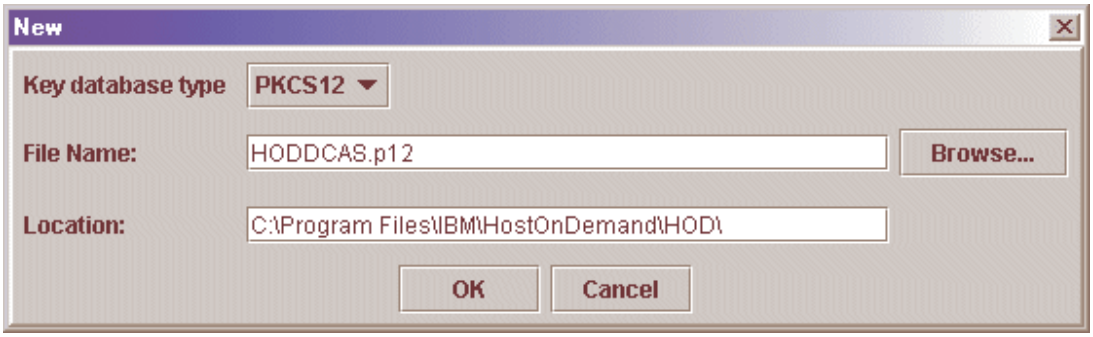
Once you have configured the the key database file to store all needed certificates, you will identify it to the CMS by specifying the name of the file as the value for the CMPI_DCAS_KEYRING_FILE parameter in Step 6: Configure the Credential Mapper Servlet on page 30.

To create the keyring database file, use the Host On-Demand Certificate Management GUI on Windows and AIX platforms, or use a P12 keyring tool for other platforms. If you plan to use a P12 keyring tool, refer the the Planning, Installing, and Configuring Host On-Demand guide in the Host On-Demand V8 InfoCenter at
http://www.ibm.com/software/webservers/hostondemand/library/v8infocenter/

In this scenario, the administrator takes the following steps to create a file called HODDCAS.p12 file on a Windows machine:

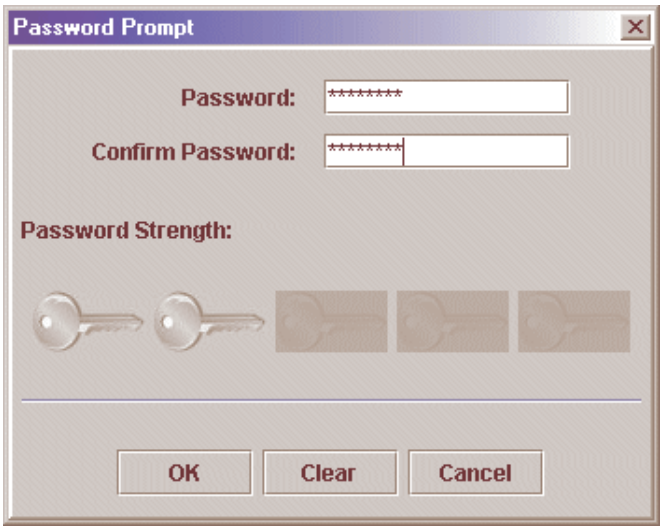
A. Click Start > Programs > IBM WebSphere Host On-Demand > Administration > Certificate Management.

B. Click Key Database File and select New. For the Key database type, select PKCS12. In the File Name field, type HODDCAS.p12. In the Location field, type C:\Program Files\IBM\HostOnDemand\HOD. Click OK.



You may chose a different name and location if desired.

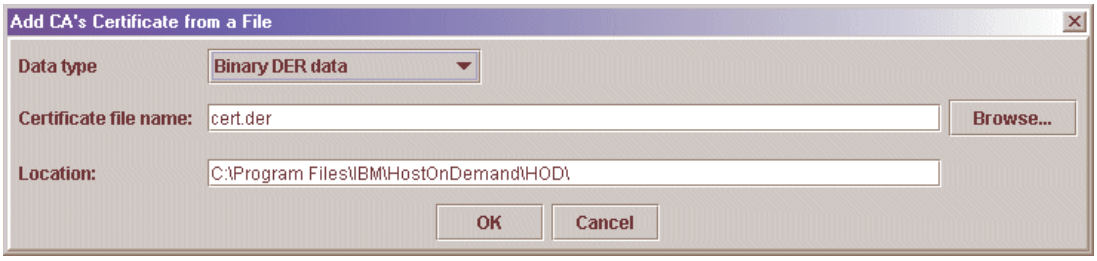
C. Type the password in the Password and Confirm Password fields. Make a note of the password. This is the password that you will use as the parameter value of the CMPI_DCAS_KEYRING_PASSWORD parameter. Click OK.



D. The next window allows you to add the DCAS server's certificate to the key database. Be sure that Key database content is set for Signer Certificates. If it is not, select the pull-down menu and change it. Click Add on the right side of the window.

E. Select Binary DER data for the data type. If the server certificate is in ASCII format, select Base64-encoded ASCII data.

F. Type the file name in the Certificate file name field, and type the path name in the Location field. Click OK.

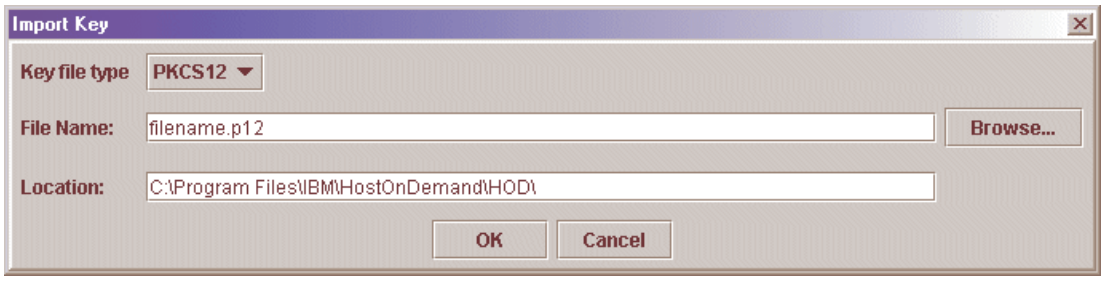


G. Enter a label for the certificate and click OK.

H. Add the DCAS client's certificate to the key database.

I. Change the Key database content to Personal Certificates and click Export/Import.

J. Select PKCS12 for the Key file type. Type the client certificate's p12 file name in the File Name field and the path name in the Location field. Click OK and enter the client certificate PIN.



K. Click OK and exit the Certificate Management GUI.

Step 5 of 11: Create the Host Credential Mapper database

The Host Credential Mapper (HCM) is one of the key players in the Web Express Logon process because it maps users' network IDs to their host IDs. Since the DCAS parameters supplied with Web Express Logon are designed to work with a JDBC database, the administrator in this scenario uses IBM DB2 to configure the HCM database. Using this type of network-accessible database provides a flexible means of associating users' network IDs with their host IDs.

Using DB2, the administrator creates a table with column names that correspond to the DCAS parameters that he will add in the next step. The following four column names are in all upper case and must exactly match the parameter values he will specify in the servlet configuration file:

- NETWORKID: This column contains the network IDs of the users. A user's network ID is the credential that uniquely identifies the user to the network security application (in this case, Tivoli Access Manager).
- HOSTADDRESS: The column contains the destination host address. This address can either be the host's IP address or the fully qualified URL, for example, amin.raleigh.ibm.com.
- APPLICATIONID: This column contains the application IDs of the users. Application IDs are used to map users' host IDs and to retrieve passtickets from the RACF server.
- HOSTID: This column contains the users' host IDs. A host ID is the credential used to uniquely identify the user to the host being accessed.

For more information about IBM DB2, refer to the following Web site: http://www.ibm.com/software/data/db2/library/.

Step 6 of 11: Configure the Credential Mapper Servlet

In this step, you will configure the Credential Mapper Servlet (CMS). The CMS is supplied with Host On-Demand and must be deployed to a J2EE-compliant Web application server. At a high level, the CMS is responsible for the following tasks: (1) determine the client's identity (called a network ID), (2) map the user's network ID to the host ID, and (3) return the host credentials to the client as an XML document.

Host On-Demand provides three CMS WAR files, one for each of the following network security applications:

- IBM Tivoli Access Manager for e-business V4.1
- Netegrity Siteminder V5.5
- Microsoft Active Directory (Windows Domain)

If you have a different network security application, you will need to customize your own version of the CMS. For more information about how to do this, refer to Customizing Web Express Logon on page 124.

In addition to several CLASS files, the WAR files contains the following four files:

- web.xml
- DCAS.xml
- Vault.xml
- was.policy

The web.xml file is the servlet configuration file that you will edit in this step. The other two XML files (DCAS.xml and Vault.xml) are sample files that we have provided to help you better understand DCAS and Vault parameters and their values. We also recommend that you use these files as a reference when you edit the web.xml file. Finally, the was.policy file is for IBM WebSphere Application Server only. It contains the required permissions for the CMS when Java 2 security is enabled. For more information, refer to Troubleshooting Web Express Logon on page 131.

In this scenario, the administrator uses WebSphere Application Server to configure the CMS. He takes the following steps:

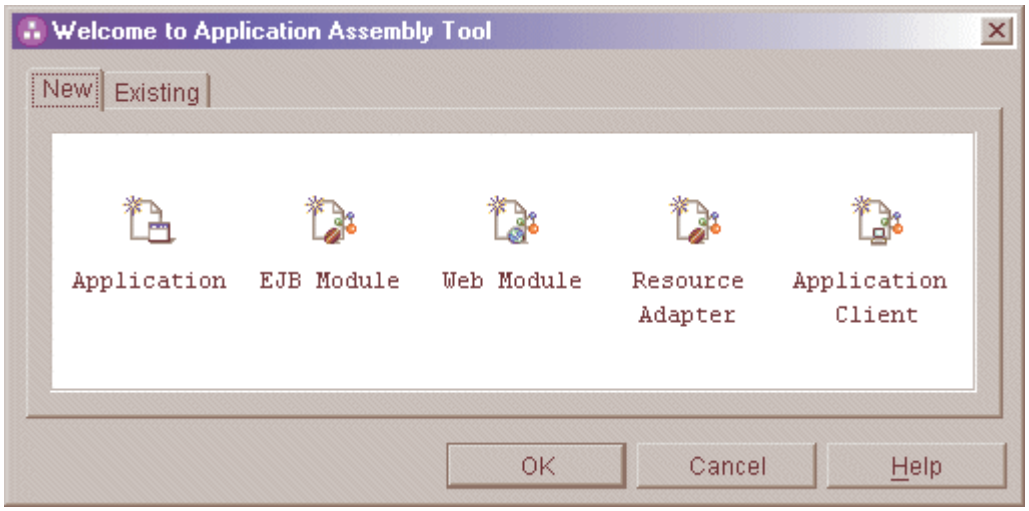
A. Locate the WAR files on the Host On-Demand CD.
B. Edit the CMS-related parameters.
C. Add optional CMS-related debugging parameters.
D. Add the required DCAS client parameters.
E. Add optional DCAS client parameters (if desired)
F. Save the WAR file.

****

A. Locate the WAR files on the Host On-Demand CD. With the Host On-Demand V8 CD loaded in the CD drive of your machine, take the following steps to locate the WAR files on the CD.

i. Click Start > Programs > IBM WebSphere > Application Server v5.0 > Application Assembly Tool.

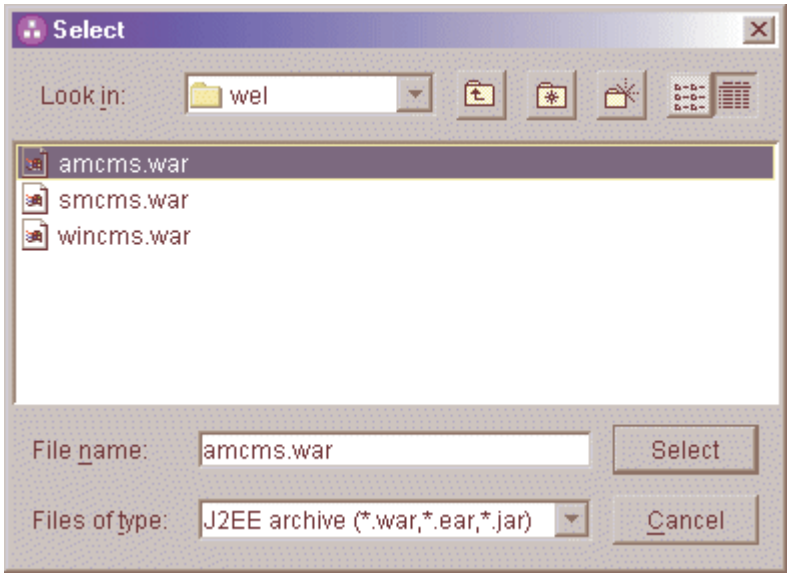
ii. On the Welcome to the Application Assembly Tool window, click the Existing tab and browse to the apps\wel directory on the Host On-Demand CD.



iii. In the apps\wel directory, you will see three WAR files, one for each of the following network security applications:

Network security application	Corresponding WAR file
IBM Tivoli Access Manager for e-business V4.1	amcms.war
Netegrity Siteminder V5.5	smcms.war
Microsoft Active Directory (Windows Domain)	wincms.war

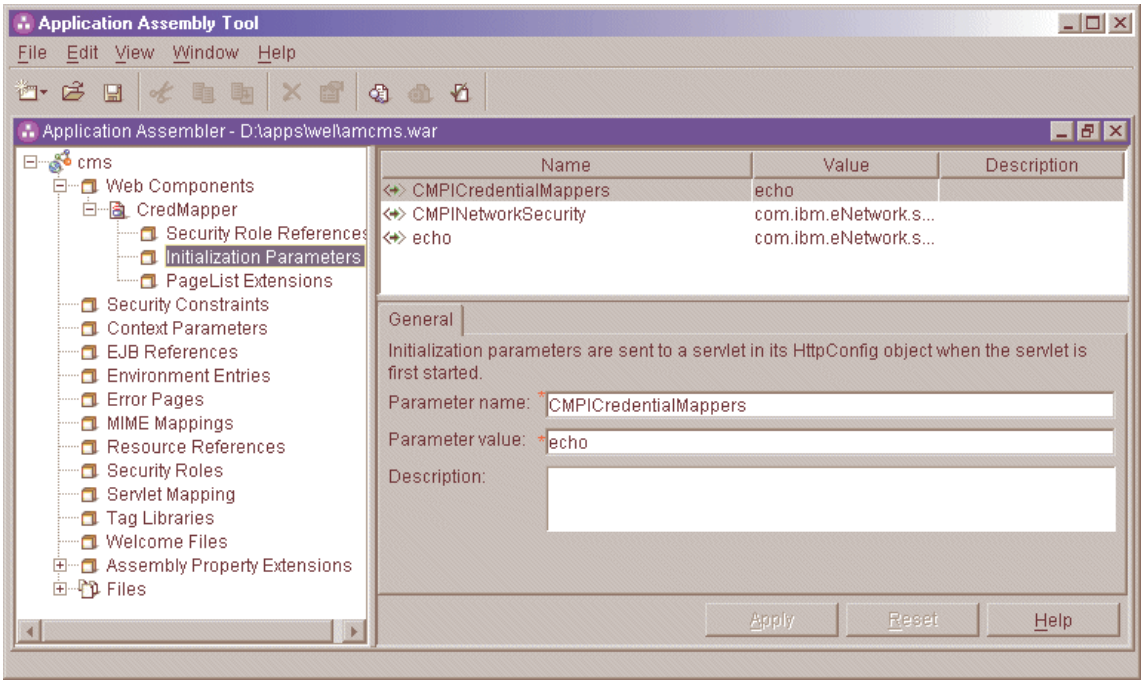
Highlight the WAR that represents your network security application and press Select. Then select OK.



If you have a different network security application, you will need to customize your own version of the CMS. For more information about how to do this, refer Customizing Web Express Logon on page 124.

B. Edit the CMS-related INIT parameters: In this step, you will edit two of the three INIT parameters in the web.xml file. You will not edit the CMPINetworkSecurity parameter name or value.

i. In the left panel of the window, expand Web Components > CredMapper and click Initialization Parameters. The three default INIT parameters specifically coded to adapt the CMS to your environment appear in the top window.



(1) - (3) describe these default parameters in more detail:

(1) Host Credential Mapper (HCM) plug-in: The name of the parameter is CMPICredentialMappers, and the parameter value is a compound value that contains the list of all available HCMs, for example, CMPIDCASPlugin and CMPIVaultPlugin. Currently, the value is echo, but you will eventually replace this with the name of your HCM plug-in.

Code example:

```
<init-param>
   <param-name>CMPICredentialMappers</param-name>
   <param-value>echo</param-value>
</init-param>
```

(2) Network Security plug-in: The name of the parameter is CMPINetworkSecurity, and the parameter value is the full path name of the class that handles the CMS interface into the network security application, which is Tivoli Access Manager in this scenario.

Code example:

```
<init-param>
   <param-name>CMPINetworkSecurity</param-name>

<param-value>com.ibm.eNetwork.security.sso.cms.CMNPIAccessManager</pa
ram-value> </init-param>
```

The Network Security plug-in does not apply to Microsoft Active Directory XML file (wincms.xml) since the Windows login ID is used as the network ID.

(3) echo plug-in: The name of this INIT parameter (echo) is the same as the value for the HCM plug-in. In a future step, you will replace echo with the name of your HCM plug-in.

Host On-Demand provides this optional echo plug-in in case you want to confirm that you are able to deploy the CMS correctly before you begin editing the web.xml file. For example, after you deploy your CMS to a Web server, you can test it by entering the following syntax in a PC's browser address bar: https://web_application_server_name/context_root/CredMapper, where web_application_server_name is the name of the Web application server, context_root is the name of the context root that you specified when deploying the CMS, and CredMapper is the name of the CMS itself.

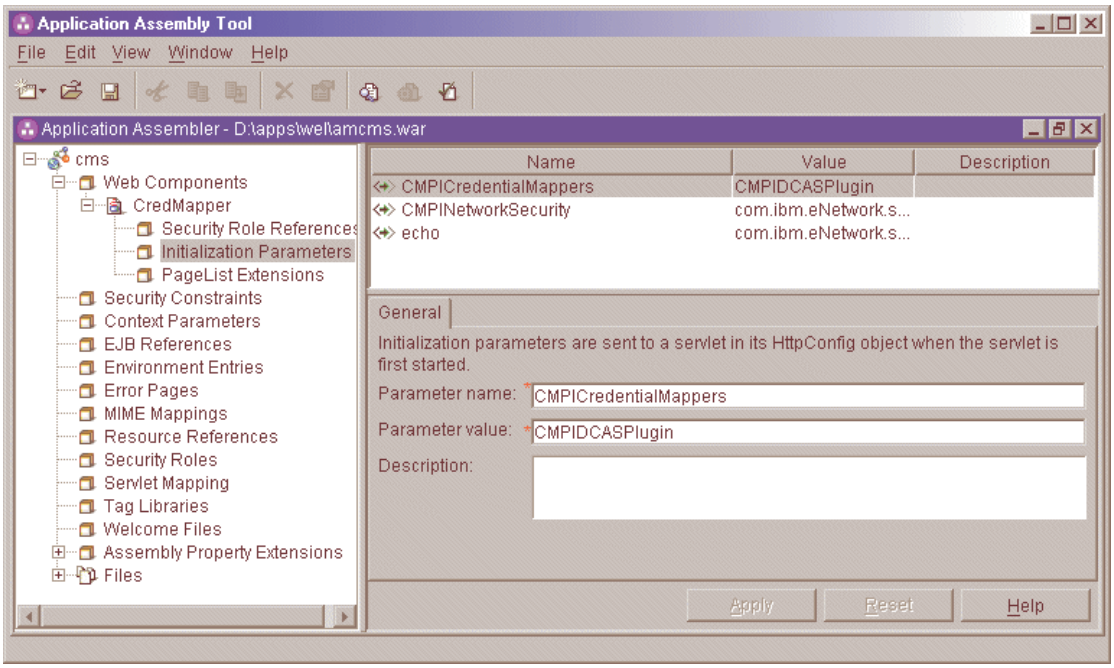
Some Web application server products allow you to deploy the servlet first and then edit the XML file. Other products, such as WebSphere Application Server V5, work best when you deploy the servlet after you edit the XML code. Refer to your product's documentation for details.

Code example:

```
<init-param>
    <param-name>echo</param-name>
          <param-value>com.ibm.eNetwork.security.sso.cms.CMPINetEcho,
                        AuthType_All,*</param-value>
</init-param>
```

ii. Highlight the CMPICredentialMappers parameter in the top panel of the window. In the Parameter value field below, change the name of its current value (echo) to the name of your HCM plug-in. In this scenario, the administrator specifies CMPIDCASPlugin as the parameter value because he is using DCAS as his HCM plug-in.

Optionally provide a description and click Apply to replace the value in the top window.



Code example:

```
<init-param>
   <param-name>CMPICredentialMappers</param-name>
   <param-value>CMPIDCASPlugin</param-value>
</init-param>
```

iii. Now highlight the echo parameter. In the Parameter name field, replace the current parameter name (echo) with the name of the parameter value that you specified for the HCM plug-in. In this scenario, the administrator changed the parameter name to CMPIDCASPlugin.

Now, replace the parameter value with a compound value that contains the full class path name of the implementing class, the authentication type to be used by the HCM plug-in, and the host mask. Separate these values with commas. In this scenario, the administrator added com.ibm.eNetwork.security.sso.cms.CMPIDCAS for the full class path name, AuthType_3270Host for the authentication type, and * for the host mask.

Full class path name
The CMS uses the value of the full class path name to create a class object of the specified type. That object is then used to handle CMS or HCM plug-in requests. The specified class file must be in the ...\WEB-INF\classes subdirectory in a loose file (not as a JAR file). From this location, the CMS will be able to access and use it whenever the need arises.

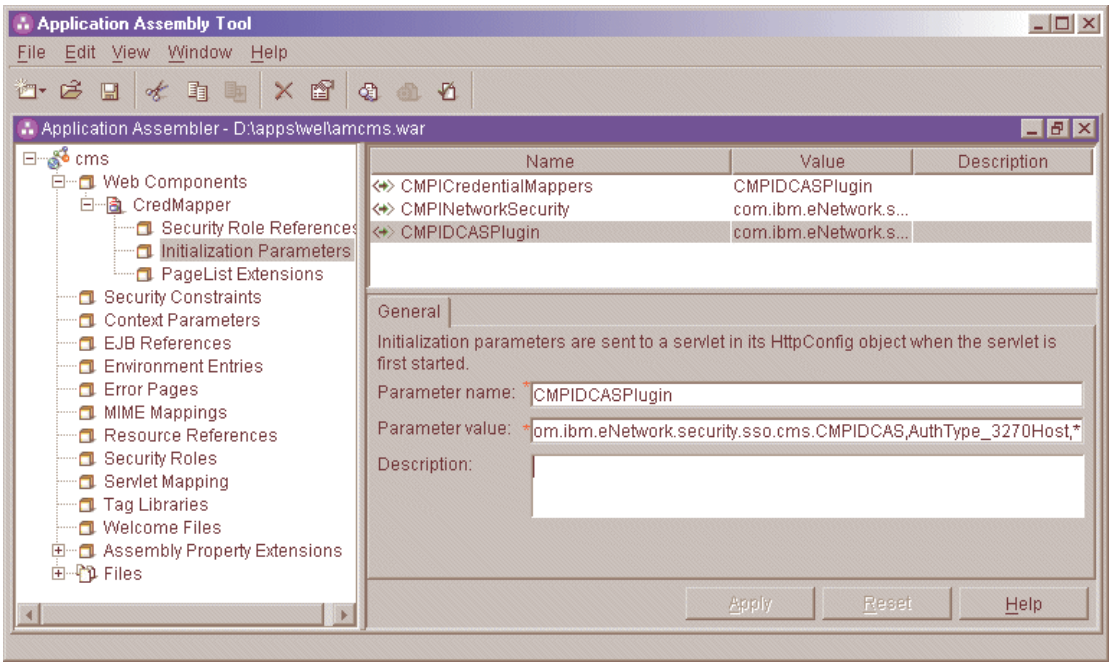
Authentication type
This parameter value is used to identify the type of authentication that the requestor needs. Once you specify the desired authentication type, the CMS can better identify which HCM plug-in to select to handle the request. You can pair multiple authentication types together to give HCM plug-ins the freedom to support multiple authentication types. Use the vertical bar character to join multiple authentication types. The six identified authentication types and descriptions are listed in the following table:

Authentication type	**Description**
AuthType_3270Host	Identifies the credentials to be used with a 3270 emulation
AuthType_5250Host	Identifies the credentials to be used with 5250 emulation
AuthType_VTHost	Identifies the credentials to be used with VT emulation
AuthType_FTPPassword	Credentials used to access an FTP host
AuthType_ConfigServer	Credentials identified by the token used to identify the user to the Host On-Demand configuration server (if you are using the Configuration server-based model)
AuthType_All	Identifies the credentials to be used for all authentication types

Host mask
The host mask is a secondary selection criteria used by the CMS to identify the most appropriate HCM plug-in. This value can contain one or more host addresses. Use the vertical bar character to join multiple addresses. Use the asterisks character to wildcard a host address. The wildcard character may start, end, or start and end a host address. The following table lists valid wild-carded addresses:

Host mask	**Value matched**
*.raleigh.ibm.com	Matches all addresses that end with .raleigh.ibm.com
ralvm*	Matches all addresses that start with ralvm
*	Matches all
xyz	Matches any host address that contains xyz

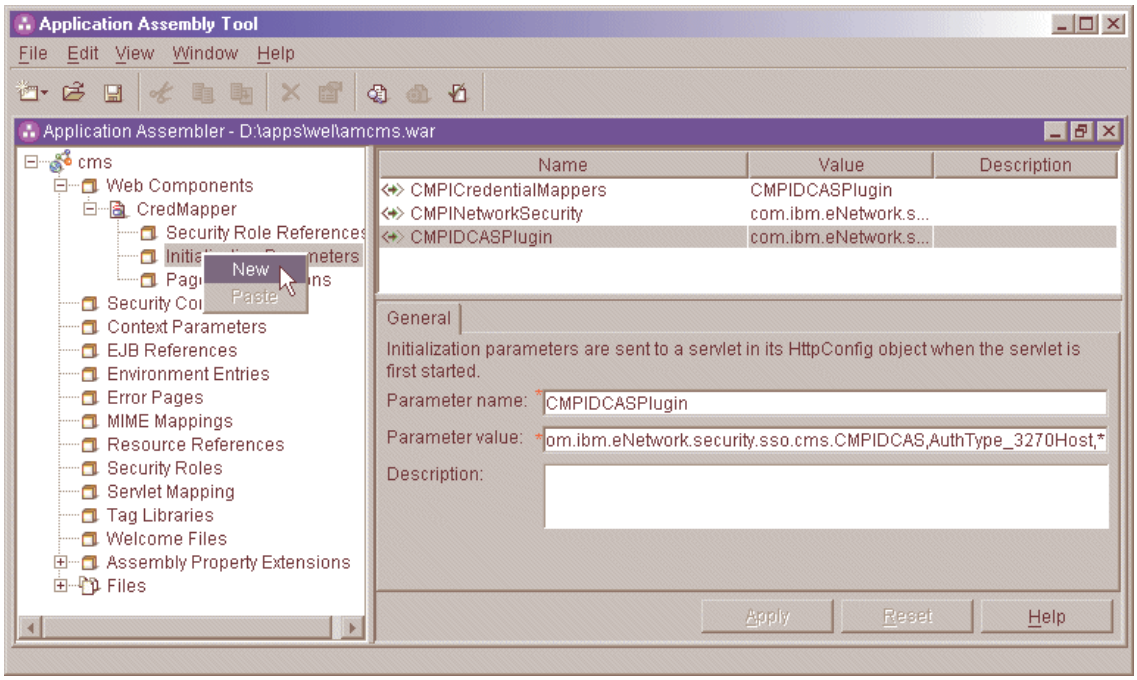
Optionally provide a description and click Apply to replace the value in the top window.



Code example:

```
<init-param>
   <param-name>CMPIDCASPlugin</param-name>
<param-value>com.ibm.eNetwork.security.sso.cms.CMPIDCAS,
   AuthType_3270Host,*</param-value>
</init-param>
```

C. Add optional CMS-related debugging parameters: To add new parameters, right-click Initialization Parameters in the left pane of the Application Assembly Tool window and select New.



Using the New Initialization Parameter window, add the following two optional debugging parameters to help you troubleshoot:

CMPI_TRACE_LOG_FILE

This parameter specifies the name of the log file. The value should be the full path to the log file, for example C:\Program Files\IBM\HostOnDemand\HOD\HODWEL.log on a Windows platform.

Code example:

```
<init-param>
   <param-name>CMPI_TRACE_LOG_FILE</param-name>
   <param-value>C:\Program
Files\IBM\HostOnDemand\HOD\HODWEL.log</param-value>
</init-param>
```

CMPI_CMS_TRACE_LEVEL

This parameter specifies the trace level for the CMS. The trace messages are logged to the log file specified by CMPI_TRACE_LOG_FILE parameter. Depending on your Web application server, they may or may not be logged to the console. Trace level values include the following:
0 = None: No tracing. This is the default.
1 = Minimum: Trace APIs and parameters, return values, and errors.
2 = Normal: Trace Minimum plus internal APIs and parameters and informational messages.
3 = Maximum: Trace Normal plus Java exceptions.

Code example:

```
<init-param>
   <param-name>CMPI_CMS_TRACE_LEVEL</param-name>
   <param-value>3</param-value>
</init-param>
```

D. Add the required DCAS client parameters for the CMPIDCASPlugin: Continue to use the New Initialization Parameter window to add the required DCAS client parameters for the CMPIDCASPlugin. Adding these parameters allows the HCM to map the user's network ID to his host ID and get a passticket from the DCAS application running on the host. The following DCAS parameters are required in order for Web Express Logon to function properly. This section is divided into three subsections, i-iii.

i. Add the following two HCM parameters to allow the client to connect to the DCAS securely:

CMPI_DCAS_KEYRING_FILE

This parameter references an SSL keyring database file that provides access to the DCAS client certificate as well as the DCAS server's certificate. The certificates establish a client-authenticated secure connection with the DCAS server. The DCAS plug-in serves as the DCAS client. You will create a keyring database file called HODDCAS.p12 in a future step.

Code example:

```
<init-param>
    <param-name>CMPI_DCAS_KEYRING_FILE</param-name>
    <param-value>C:\Program
Files\IBM\HostOnDemand\HOD\HODDCAS.p12</param-value>
</init-param>
```

CMPI_DCAS_KEYRING_PASSWORD

This parameter specifies the password for the keyring database.

Code example:

```
<init-param>
     <param-name>CMPI_DCAS_KEYRING_PASSWORD</param-name>
<param-value>45ie8WciVu</param-value>
</init-param>
```

We strongly recommend that you encrypt this parameter using the password encryption tool provided with Host On-Demand. The tool encrypts the password and then decrypts it so the HCM can use it. To learn more about how to use this tool, refer to Password Encryption Tool on page 140.

ii. The following parameters contain all the relevant information needed to connect to your HCM, which in this case is a JDBC database table created with IBM DB2. You can either configure access to an existing database or to a newly created one. The level of security for the database depends on the database vendor. See the documentation for details.

CMPI_DCAS_DB_ADDRESS

This is a URL string that provides the address of the database. An example of this string is jdbc:db2://dtagw.raleigh.ibm.com:6789/HODSSO.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_DB_ADDRESS</param-name>
   <param-value>jdbc:db2://dtagw.raleigh.ibm.com:6789/
      HODSSO</param-value>
</init-param>
```

CMPI_DCAS_DB_NET_DRIVER

This string contains the name of the class that acts as the network database driver. An example of this string is COM.ibm.db2.jdbc.net.DB2Driver. The location of this class is assumed to be in the existing class path.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_DB_NET_DRIVER</param-name>
   <param-value>COM.ibm.db2.jdbc.net.DB2Driver</param-value>
</init-param>
```

CMPI_DCAS_DB_USERID

This is the ID of the user account to use when accessing the database. In this case, the user ID is admin.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_DB_USERID</param-name>
   <param-value>admin</param-value>
</init-param>
```

CMPI_DCAS_DB_PASSWORD

This is the password of the user account to use when accessing the database. We strongly recommend that you encrypt this parameter using the password encryption tool provided with Host On-Demand. The tool encrypts the password and then decrypts it so the HCM plug-in can use it. To learn more about how to use this tool, refer to Password Encryption Tool on page 140.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_DB_PASSWORD</param-name>
   <param-value>tuBu9v8lHiJi1jt08UgHzA==</param-value>
</init-param>
```

CMPI_DCAS_DB_TABLE

This identifies the table to use for the needed query. In this case, the table is called HACP.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_DB_TABLE</param-name>
   <param-value>HACP</param-value>
</init-param>
```

iii. The following parameters correspond directly to the column names that you added to your HCM database table in Step 5: Create the Host Credential Mapper database. Recall that you added the following four column names, all in uppercase: NETWORKID, HOSTADDRESS, APPLICATIONID, and HOSTID.

Based on the information provided by the first three of these parameters (network ID, host address, and the host application ID), you can make a SQL query of the database to get the host ID. The result of the query is entered in the host ID (HOSTID) column. Assuming that the query is successful, a call is made to the DCAS to request the passticket.

CMPI_DCAS_DB_NETID_COL_NAME
This entry identifies the name of the column that contains the network ID value (NETWORKID).

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_DB_NETID_COL_NAME</param-name>
   <param-value>NETWORKID</param-value>
</init-param>
```

CMPI_DCAS_DB_HOSTADDR_COL_NAME
This entry identifies the name of the column that contains the host address value (HOSTADDRESS).

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_DB_HOSTADDR_COL_NAME</param-name>
   <param-value>HOSTADDRESS</param-value>
</init-param>
```

CMPI_DCAS_DB_HOSTAPP_COL_NAME
This entry identifies the name of the column that contains the host application value (APPLICATIONID).

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_DB_HOSTAPP_COL_NAME</param-name>
   <param-value>APPLICATIONID</param-value>
</init-param>
```

CMPI_DCAS_DB_HOSTID_COL_NAME
This entry identifies the name of the column that contains the host ID value (HOSTID).

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_DB_HOSTID_COL_NAME</param-name>
   <param-value>HOSTID</param-value>
</init-param>
```

E. Add optional DCAS client parameters (if desired): Unlike the previous set of DCAS parameters, the following parameters are optional. Which of these parameters you add to the web.xml file depends on your environment and your objectives as an administrator:

CMPI_DCAS_TRACE_LEVEL
This parameter specifies the trace level for the DCAS plug-in. The trace messages are logged to the log file specified by CMPI_TRACE_LOG_FILE parameter. Depending on your Web application server, they may or may not be logged to the console. Trace level values include the following:
0 = None: No tracing. This is the default.
1 = Minimum: Trace APIs and parameters, return values, and errors.
2 = Normal: Trace Minimum plus internal APIs and parameters and informational messages.
3 = Maximum: Trace Normal plus Java exceptions.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_TRACE_LEVEL</param-name>
   <param-value>3</param-value>
</init-param>
```

CMPI_DCAS_HOST_PORT

The DCAS host address is determined based on the destination host specified in the request. The default port address of 8990 is used, but you may override it using this parameter.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_HOST_PORT</param-name>
   <param-value>8990</param-value>
</init-param>
```

CMPI_DCAS_USE_WELLKNOWN_KEYS

This parameter indicates whether the WellKnownTrustedCAs.class should be used to look up the DCAS server certificate or not. The WellKnownTrustedCAs.class file must be in the root directory of the CMS. The default is true.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_USE_WELLKNOWN_KEYS</param-name>
   <param-value>true</param-value>
</init-param>
```

CMPI_DCAS_VERIFY_SERVER_NAME

This parameter indicates if the server host name in the certificate must be verified in addition to the certificate validation. The default is false.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_VERIFY_SERVER_NAME</param-name>
   <param-value>false</param-value>
</init-param>
```

CMPI_DCAS_REQUEST_TIMEOUT

This parameter specifies the passticket request timeout in milliseconds. It should be less than the Host On-Demand macro time-out value. The default is 50000.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_REQUEST_TIMEOUT</param-name>
   <param-value>50000</param-value>
</init-param>
```

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_REQUEST_TIMEOUT</param-name>
   <param-value>50000</param-value>
</init-param>
```

CMPI_DCAS_DB_PRESERVE_WHITESPACE

This parameter indicates whether to trim white spaces from the credential request parameters or not. If true, the white spaces are not trimmed. The default is false.

Code example:

```
<init-param>
   <param-name>CMPI_DCAS_DB_PRESERVE_WHITESPACE</param-name>
   <param-value>false</param-value>
</init-param>
```

F. Save the WAR file: On the Application Assembly Tool window, click File > Save As to save your WAR file to your preferred location. If it saves successfully, this window will appear as a confirmation:

The following is the administrator's completed XML file after editing and adding the necessary parameters:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">

<web-app id="WebApp_ID">
    <display-name>cms</display-name>
<description>Credential Mapper Servlet</description>

<servlet>
    <servlet-name>CredMapper</servlet-name>
    <display-name>CredMapper</display-name>
<servlet-class>com.ibm.eNetwork.security.sso.cms.CredMapper</servlet-class>

<init-param>
    <param-name>CMPICredentialMappers</param-name>
    <param-value>CMPIDCASPlugin </param-value>
</init-param>

<init-param>
    <param-name>CMPINetworkSecurity</param-name>
    <param-value>com.ibm.eNetwork.security.sso.cms.CMNPIAccessManager
    </param-value>
</init-param>

<init-param>
    <param-name>CMPIDCASPlugin</param-name>
    <param-value>com.ibm.eNetwork.security.sso.cms.CMPIDCAS,AuthType_3270Host,*</para
    m-value>
</init-param>

<init-param>
    <param-name>CMPI_TRACE_LOG_FILE</param-name>
    <param-value>C:\Program Files\IBM\HostOnDemand\HOD\HODWEL.log</param-value>
    <description>Credential Mapper Log file name.</description>
</init-param>

<init-param>
    <param-name>CMPI_CMS_TRACE_LEVEL</param-name>
    <param-value>3</param-value>
    <description>DCAS Trace level. 0=none,1=min,2=normal,3=max.</description>
</init-param>

<init-param>
    <param-name>CMPI_DCAS_KEYRING_FILE</param-name>
    <param-value>C:\Program Files\IBM\HostOnDemand\HOD\HODDCAS.p12</param-value>
    <description>An SSL key database file that contains the client and server
    certificate information.</description>
</init-param>

<init-param>
    <param-name>CMPI_DCAS_KEYRING_PASSWORD</param-name>
    <param-value>45ie8WciVu=</param-value>
```

```
    <description>Key database file password. Use the encrypt password tool.
    </description>
</init-param>

<init-param>
    <param-name>CMPI_DCAS_DB_ADDRESS</param-name>
    <param-value>jdbc:db2://bhttd.raleigh.ibm.com:6789/HODSSO</param-value>
    <description>This is a URL string that provides the address
    of the database.</description>
</init-param>

<init-param>
    <param-name>CMPI_DCAS_DB_NET_DRIVER</param-name>
    <param-value>COM.ibm.db2.jdbc.net.DB2Driver</param-value>
    <description>This string contains the name of the class that will act as the network database
    driver. The location of this class is assumed to be in the existing classpath.</description>
</init-param>

<init-param>
    <param-name>CMPI_DCAS_DB_USERID</param-name>
    <param-value>admin</param-value>
    <description>This is the identification of the user account to use when accessing the
    database.</description>
</init-param>

<init-param>
    <param-name>CMPI_DCAS_DB_PASSWORD</param-name>
    <param-value>tuBu9v8lHiJi1jt08UgHzA==</param-value>
    <description>This is the password of the user account to use when accessing the
    database.</description>
</init-param>

<init-param>
    <param-name>CMPI_DCAS_DB_TABLE</param-name>
    <param-value>HACP</param-value>
    <description>This entry identifies the table to use for the needed query.</description>
</init-param>

<init-param>
    <param-name>CMPI_DCAS_DB_NETID_COL_NAME</param-name>
    <param-value>NETWORKID</param-value>
    <description>Column name that contains the Network ID value</description>
</init-param>

<init-param>
    <param-name>CMPI_DCAS_DB_HOSTADDR_COL_NAME</param-name>
    <param-value>HOSTADDRESS</param-value>
    <description>Column name that contains host address value</description>
</init-param>

<init-param>
    <param-name>CMPI_DCAS_DB_HOSTAPP_COL_NAME</param-name>
    <param-value>APPLICATIONID</param-value>
    <description>Column name that contains host application value</description>
</init-param>
```

```
<init-param>
    <param-name>CMPI_DCAS_DB_HOSTID_COL_NAME</param-name>
    <param-value>HOSTID</param-value>
    <description>Column name that contains host user ID value</description>
</init-param>

</servlet>
<servlet-mapping>
<servlet-name>CredMapper</servlet-name>
<url-pattern>/CredMapper</url-pattern>
</servlet-mapping>
</web-app>
```

Step 7 of 11: Deploy the Credential Mapper Servlet.

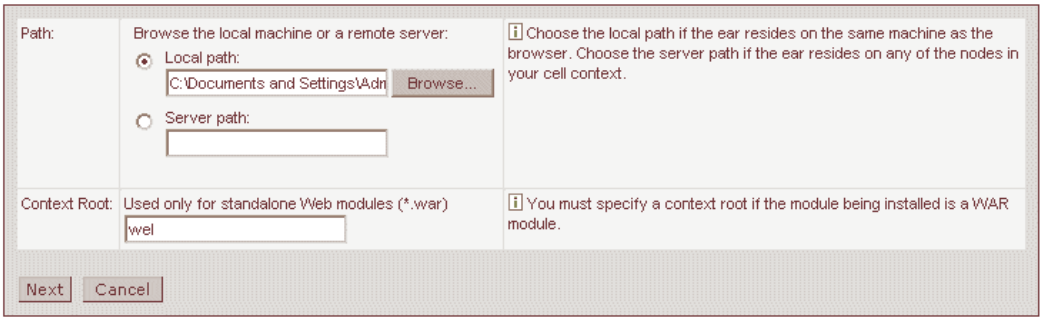
The way in which you deploy the CMS depends on your Web application server. In this scenario, the administrator uses WebSphere Application Server V5 on a Windows platform to deploy the CMS. He takes the following steps:

A. Click Start > Programs > IBM WebSphere > Application Server v5.0 > Administrative Console to open the Administrative Console.

B. In the left navigation field, click Applications > Install New Application.

C. Select Local path and browse to the amcms.war file.

D. Specify the content root in the Context Root field. The context root is combined with the defined servlet name (CredMapper) to compose the full URL that users type to access the servlet. For example, if the context root is /wel , then the URL is https://host:port/wel/CredMapper, where host is the name of the host, port is the port number, and CredMapper is the name of the CMS. Click Next.



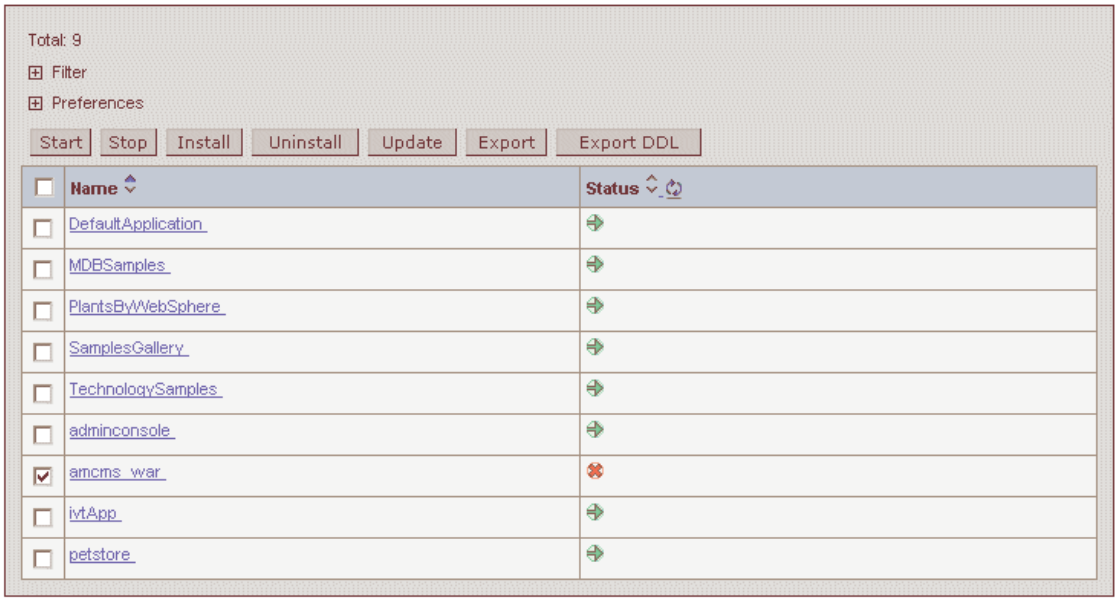
E. Check the Generate Default Bindings box. By choosing this option, you can jump directly to the Summary step and deploy the WAR file. Click Next.

F. Scroll down and click the Step 4 Summary link.

G. Scroll down and click Finish.

H. In the left navigation field, click Applications > Enterprise Applications.

I. Scroll through the applications and check the checkbox beside amcms_war, which is your application name. Notice that the status is Stop until you start it in the next step.



J. Save the application and click Start at the top to start the application. Once the application starts, the status will change from the Stop icon,

to the Start icon.

K. Test to make sure that the WAR file installed correctly by pointing a browser to http://server_name:9080/wel/CredMapper, where server_name is the name of the host server. Port 9080 is the default WebSphere port. If you deployed the servlet correctly, your browser request will return XML code. If this fails, try stopping and restarting the application again.

Step 8 of 11: Begin creating your HTML file.

The Host On-Demand Deployment Wizard allows you to create an HTML file that is used to launch Host On-Demand sessions. Within the Deployment Wizard, you can add, delete, configure, and start sessions. It guides you configuration choices and provides comprehensive help for the features. When you have finished selecting features, it creates the HTML and supporting files for you.

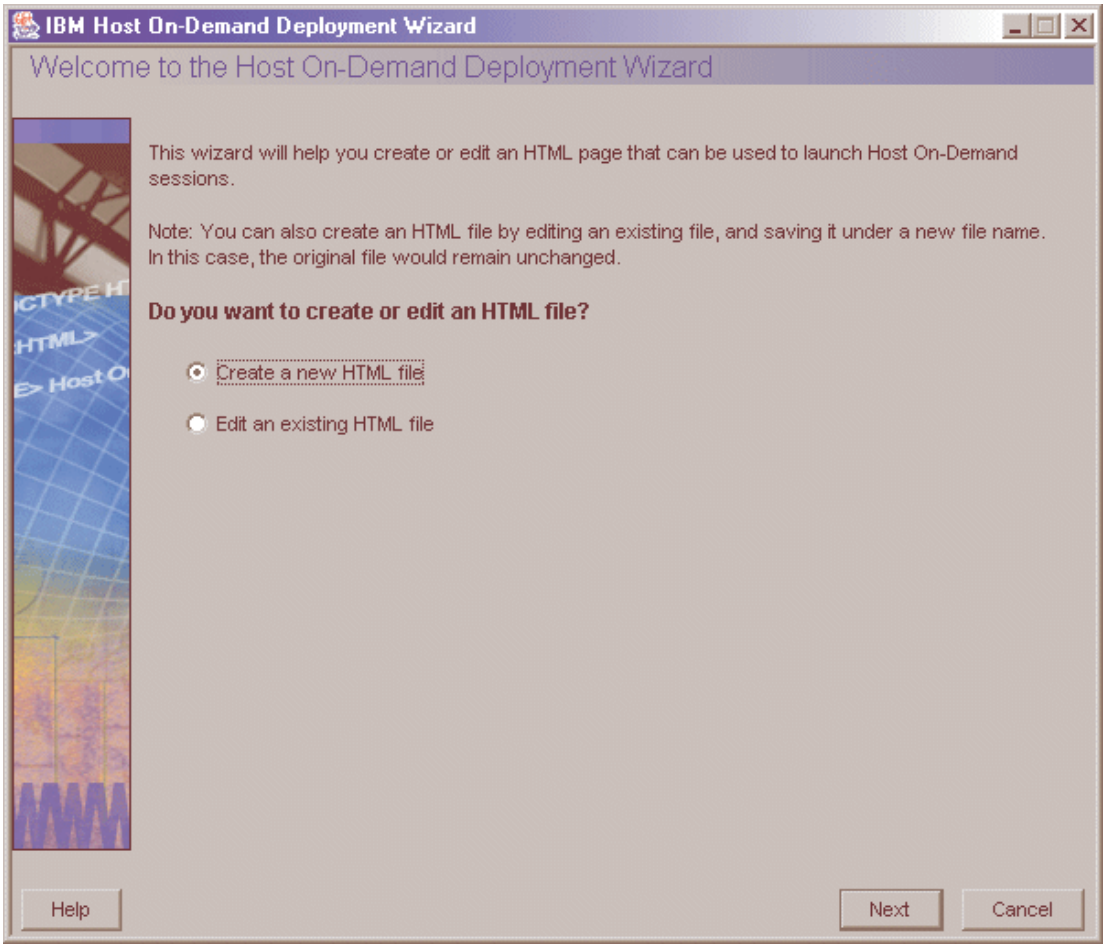
In this scenario, the administrator performs Steps 8 - 11 all within the Deployment Wizard in one sitting. However, you may decide to create your HTML file first and then configure your session and create your macro later. Refer to the supplemental notes at the beginning of Steps 9 and 10 for more information about how to do this.

To begin creating your HTML file on a Windows machine, take the following steps:

A. Open the Deployment Wizard:

- If you automatically installed the Deployment Wizard as part of the Windows Host On-Demand server, click Start > Programs > IBM WebSphere Host On-Demand > Administration > Deployment Wizard.
- If you installed the Deployment Wizard from the Host On-Demand CD separately, click Start > Programs > IBM WebSphere Host On-Demand Deployment Wizard > Deployment Wizard.

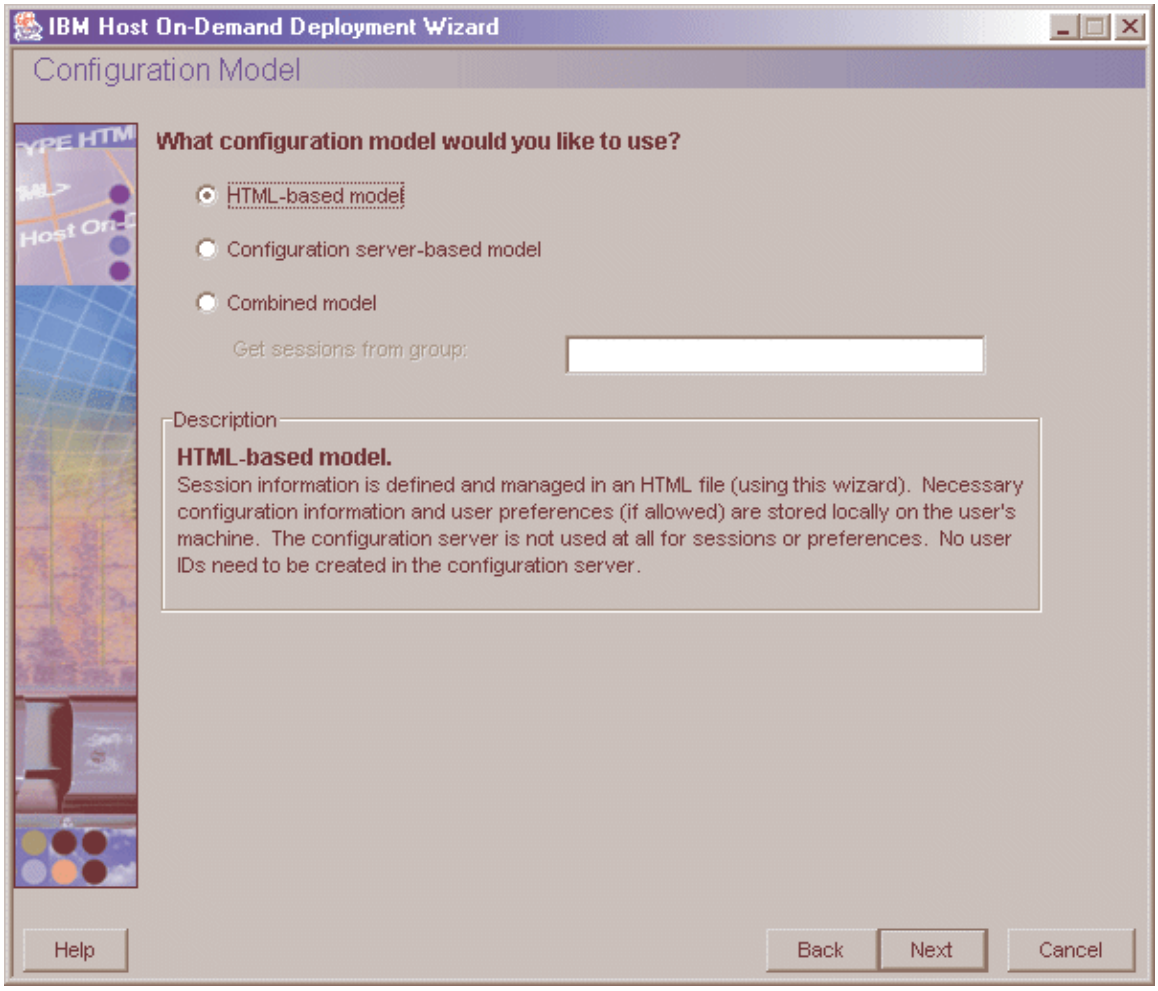
B. Select either to create a new HTML file or edit an existing file. Click Next.



C. Select from the following three configuration models and click Next:

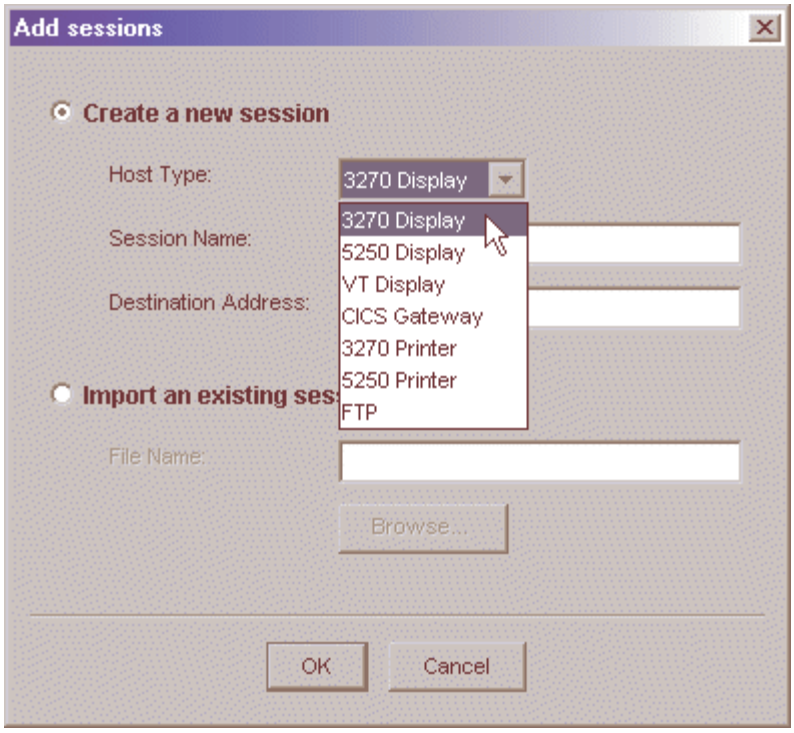
- HTML-based model
- Configuration server-based model *(*if you choose this configuration model, refer to 121.)
- Combined model

The administrator in this scenario selects the HTML-based model.



D. On the Host Sessions window, click New/Import to open the Add sessions window. This window allows you to either create a new session (default) or import an existing session.

To create a new session, select a host type, enter a session name, and a destination address. In this scenario, the administrator selects 3270 Display.



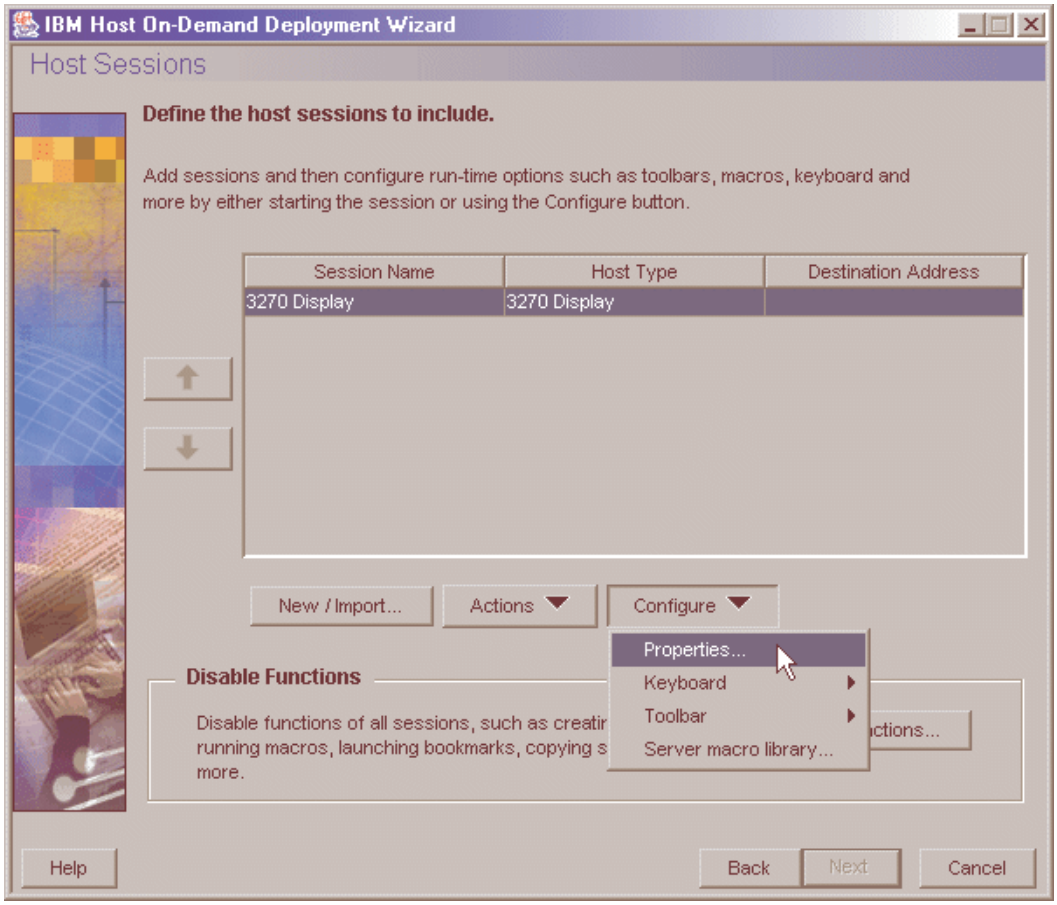
Click OK to return to the Host Sessions window.

Step 9 of 11: Configure your Host On-Demand session.

In this step, you will configure your Host On-Demand session to use Web Express Logon. At this point, you are still using the Deployment Wizard tool (continued from Step 8).

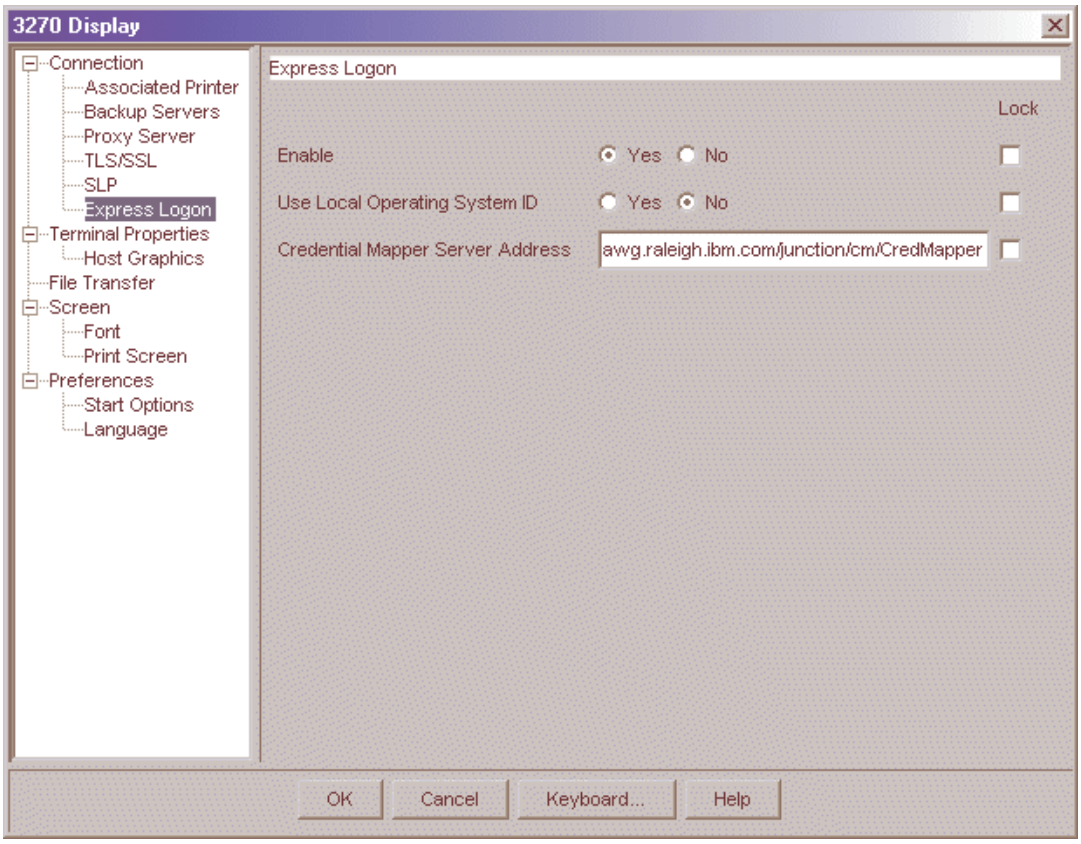
If you have already created your HTML file and now wish to configure it to use Web Express Logon, open the Host On-Demand desktop, right-click the session icon, and select Properties. Skip to Step B.

A. On the Host Sessions window, click Configure > Properties to configure your session to use Web Express Logon.



B. Under the Connection option on the left side of the window, click Express Logon. Select Yes to enable Express Logon and chose whether or not you want Host On-Demand to use the user's local operating system ID for authentication. Next, type the full URL of the credential mapper server, for example, https://*server_name*/*junction*/*cm*/*CredMapper*, where

- *server_name* is the name of the authentication server
- *junction* is the name of the junction point (optional)
- *cm* is the credential mapper servlet space
- *CredMapper* is the servlet name



Be sure that the servlet name matches the name in your XML file. For example, if you specify the servlet name in your host session as CredMapper (recommended), the code in your XML should look like the following:

```
   <servlet>
      <servlet-name>CredMapper</servlet-name>
      <display-name>CredMapper</display-name>
<servlet-class>com.ibm.eNetwork.security.sso.cms.CredMapper</servlet-cla
ss>
```

The servlet that resides at this URL processes the HTTPS request from the user, performs a lookup, and returns the user's credentials. The Host On-Demand client uses the obtained credentials to automate the login process.

Click OK to return to Host Sessions window.

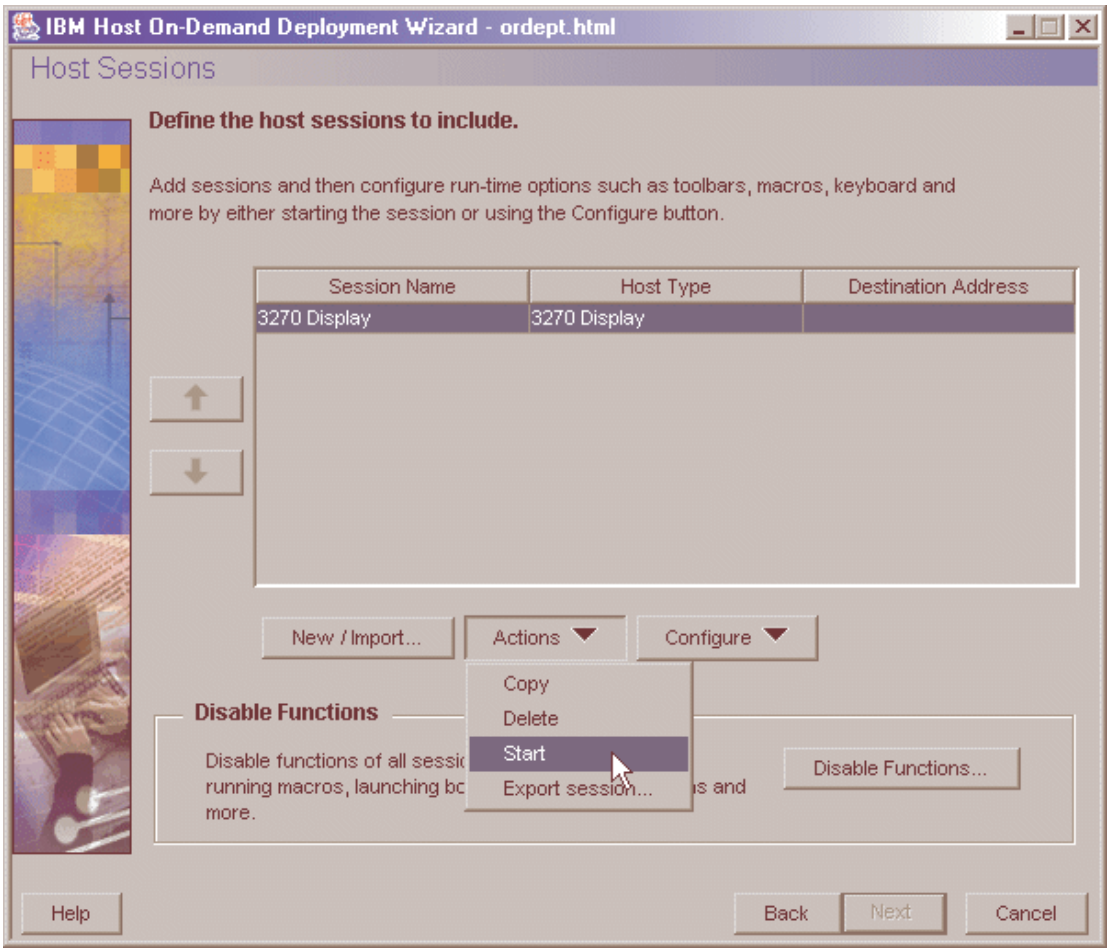
Step 10 of 11: Record the Web Express Logon macro.

In this step, you will create the component that ties the whole logon automation process together — the login macro. You record this during an active session. At this point, you are still using the Deployment Wizard tool (continued from Step 9).

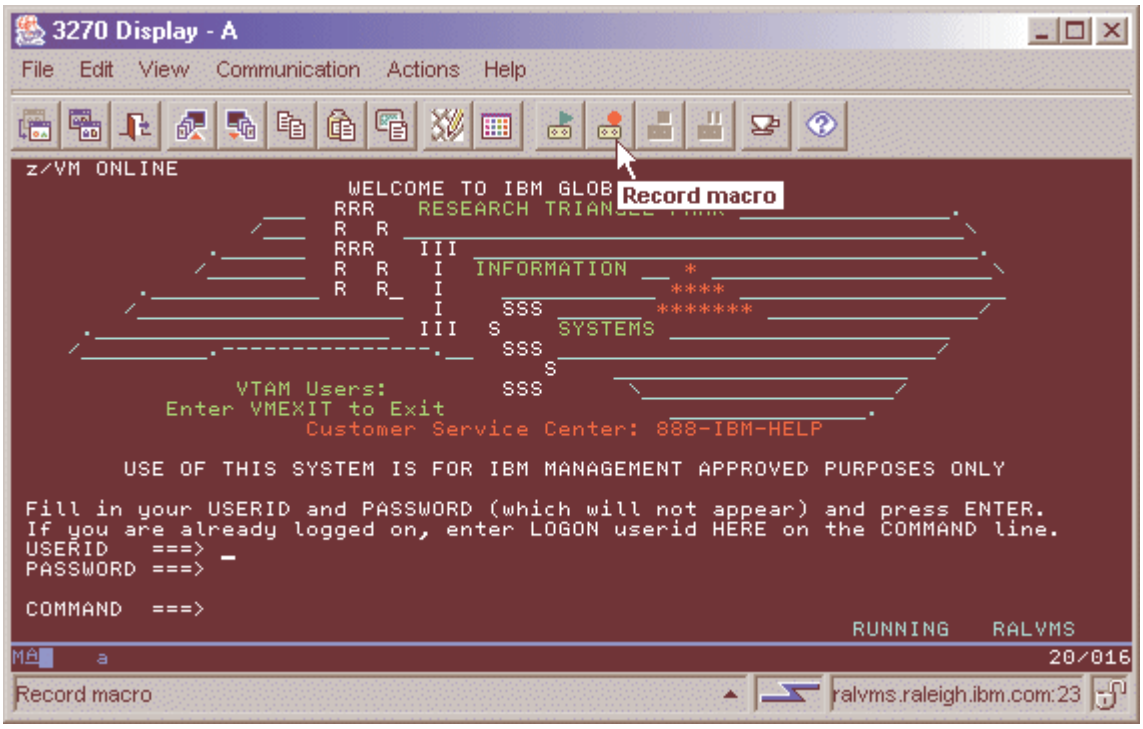
If you have already created your HTML file and now wish to record the login macro, open the Host On-Demand desktop, right-click the session icon, and select Start. Skip to Step B.

To record the macro, take the following steps:

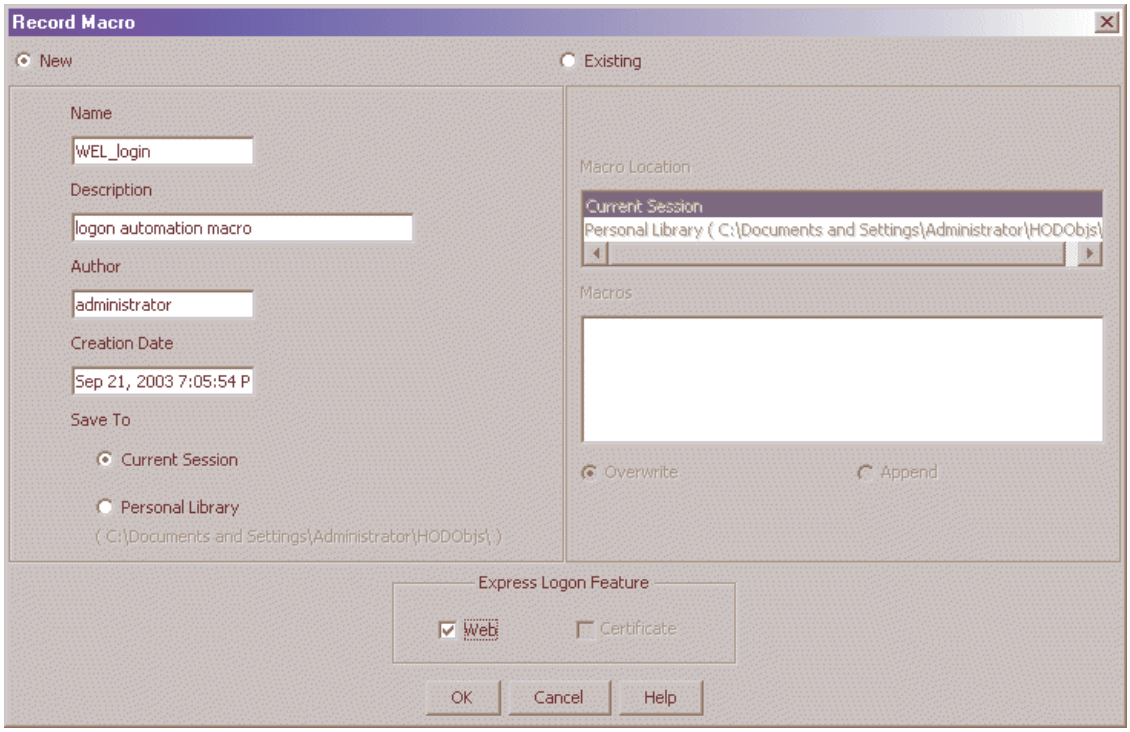
A. On the Host Sessions window, click Actions > Start to start your session.



B. Click the Record macro button on the toolbar of your active session.



C. On the Record Macro window, select New and fill in the Name and Description (optional) fields. Check Web under Express Logon Feature at the bottom of the window. Click OK.



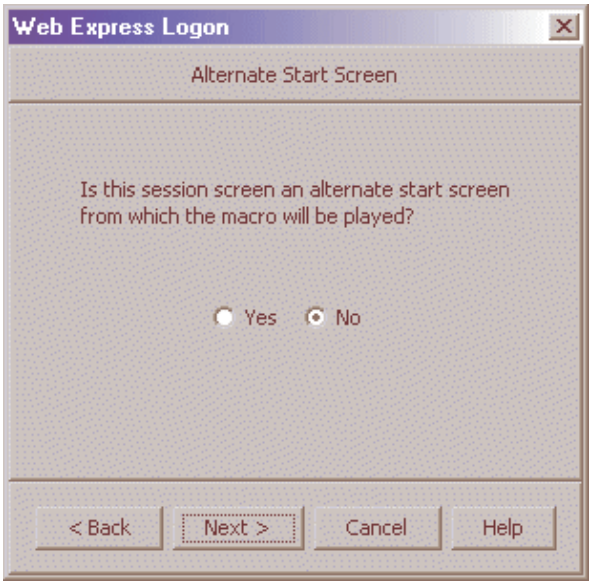
D. Enter the application ID (3270 sessions only) in the Application ID window and then click OK.

This name must match the RACF PTKTDATA (Passticket Data Profile) application name that is configured on the z/OS host. This name could be the same as the application name that the user is logging on to (for example, the name on USSMSG10). When creating PTKTDATA profiles for applications such as TSO (time sharing option), the application name portion of the profile will most likely not be the same. For example, RACF requires that the application ID portion of the profile name be TSO+SID. Refer to the RACF Security Administrator's Guide to determine the correct profile naming. You can obtain this ID from the host administrator.

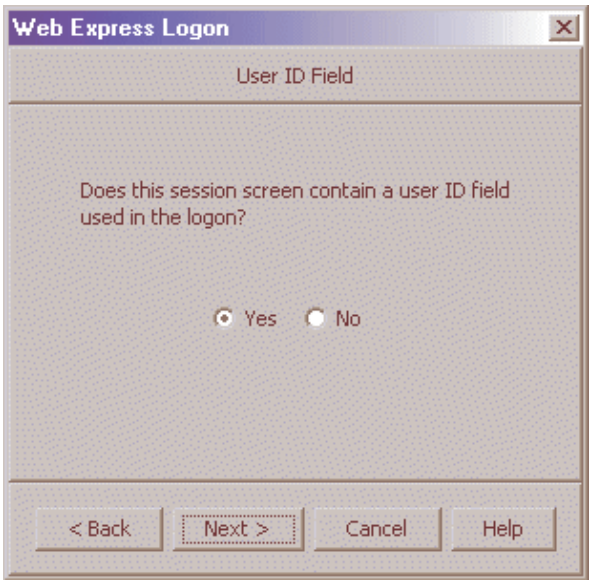
E. The Screen Criteria window shows you what is needed by the macro to complete the logon. Once you reach a screen that meets any of the criteria, click OK.

F. On the Alternate Start Screen window, specify whether this screen is an alternate start screen and click Next. The macro can start playing when a start screen is recognized or when an alternate screen is recognized. You can have only one alternate start screen per logon. If you have multiple logons, you will pass through this screen Again.

The alternate start screen is a screen from which the user might want to play the macro to log on to the application. If the application has more than one possible start screen, you should identify it during the recording process so that the macro can be played from that screen. For example, the logon process might begin from the USSMSG10 screen or the application logon screen. You may start the logon macro from either the start screen or the alternate start screen. You can designate only one screen as an alternate start screen. There is no alternate start screen after the screen that contains the user ID.

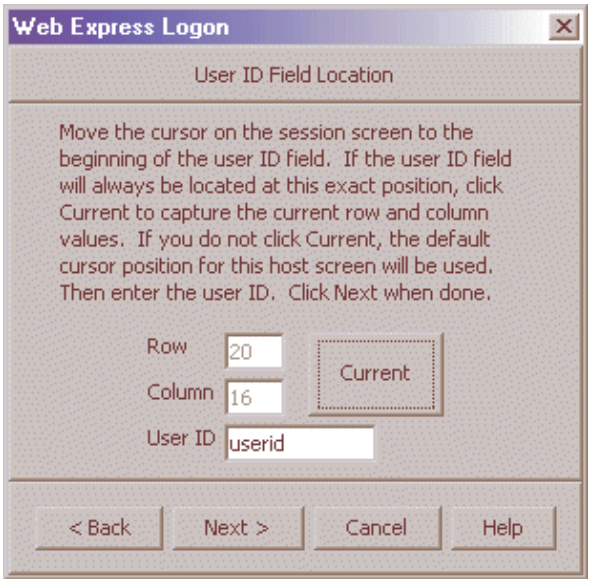


G. On the User ID Field window, select Yes to specify that the session screen contains a user ID field. Click Next.



H. On the User ID Field Location window, type the user ID in the User ID field, not on the session screen. You must enter a user ID to continue recording the macro. The macro enters the actual user ID text in the user ID field on the session screen. Row/column determines the cursor position on the screen for the user ID field. Click Current to use the cursor's current position on the session screen if you know it is correct. If the current cursor position is not correct, move the cursor to the beginning of the user ID field on the session screen to identify where the user will enter the user ID and click Current. The field values change to match the new cursor position on the screen. If the initial cursor position is correct, then there is no need to move the cursor on the session screen. When you are finished, click Next.

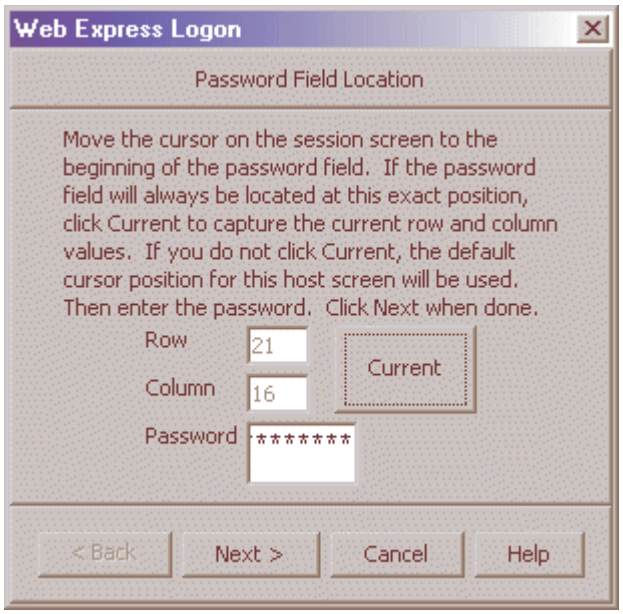
Click Current only if you will not be using this screen for multiple applications and the location of the user ID field never changes.



I. On the Password Field window, select Yes to specify that the session screen contains a password field. Click Next.

J. On the Password Field Location window, type the password in the Password field on this window, not on the screen. You must enter a password to continue recording the macro. The macro enters the actual password text in the password field on the session screen. Row/Column determines the cursor position on the screen for the password field. Click Current to use the cursor's current position on the session screen if it is correct. If not, move the cursor to the beginning of the Password field on the session screen to specify where the user will enter the password and click Current. The field values change to match the new cursor position on the screen. If the initial cursor position is correct, then there is no need to move the cursor on the session screen. When you are finished, click Next.

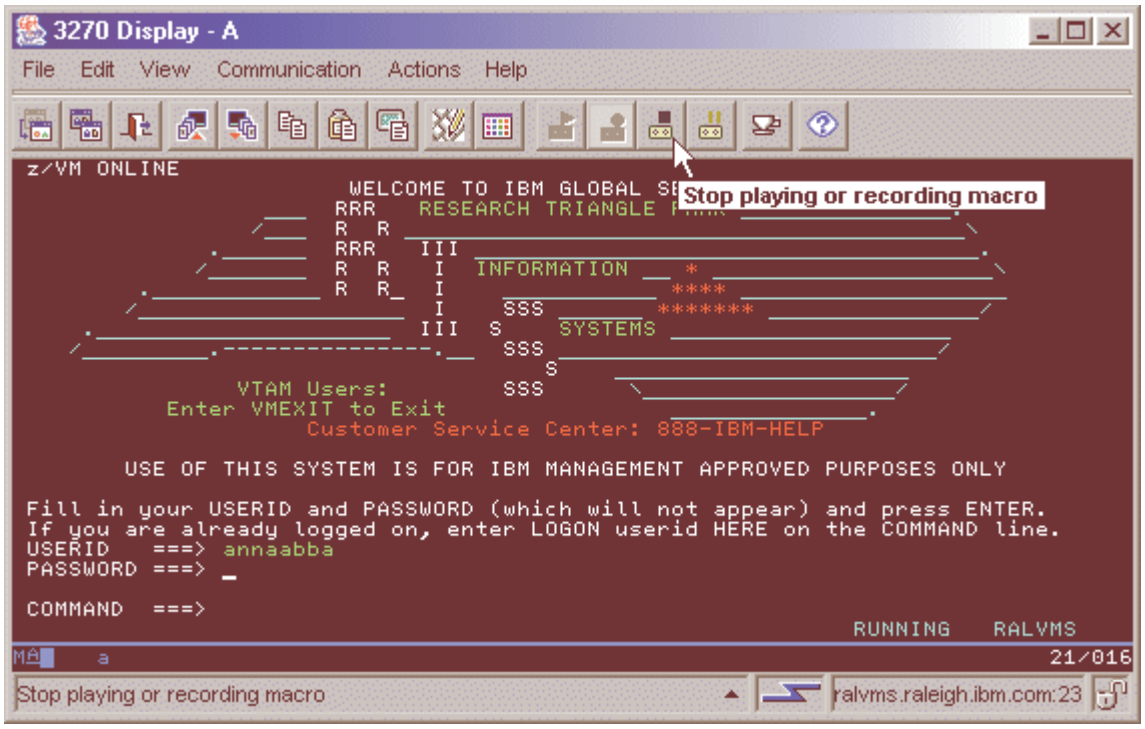
Click Current only if you will not be using this screen for multiple applications and the location of the password field never changes.



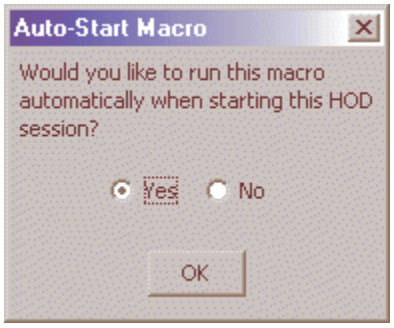
K. On the Multiple Logons window, select No and click Finish to finish recording the logon portion of the macro. Click Yes only if you you want to define another logon sequence for an additional Application.

L. Click OK to stop recording the Macro.

M. Finish recording your macro using the Macro Manager, and click the Stop recording macro button on the Toolbar.



N. If you are planning to save the macro to your current session (and not to a file), another window appears that asks you if you would like the macro to start automatically when the user opens the session. Click Yes if you would like the macro to auto-start. If you select No, the user will have to start the macro Manually.

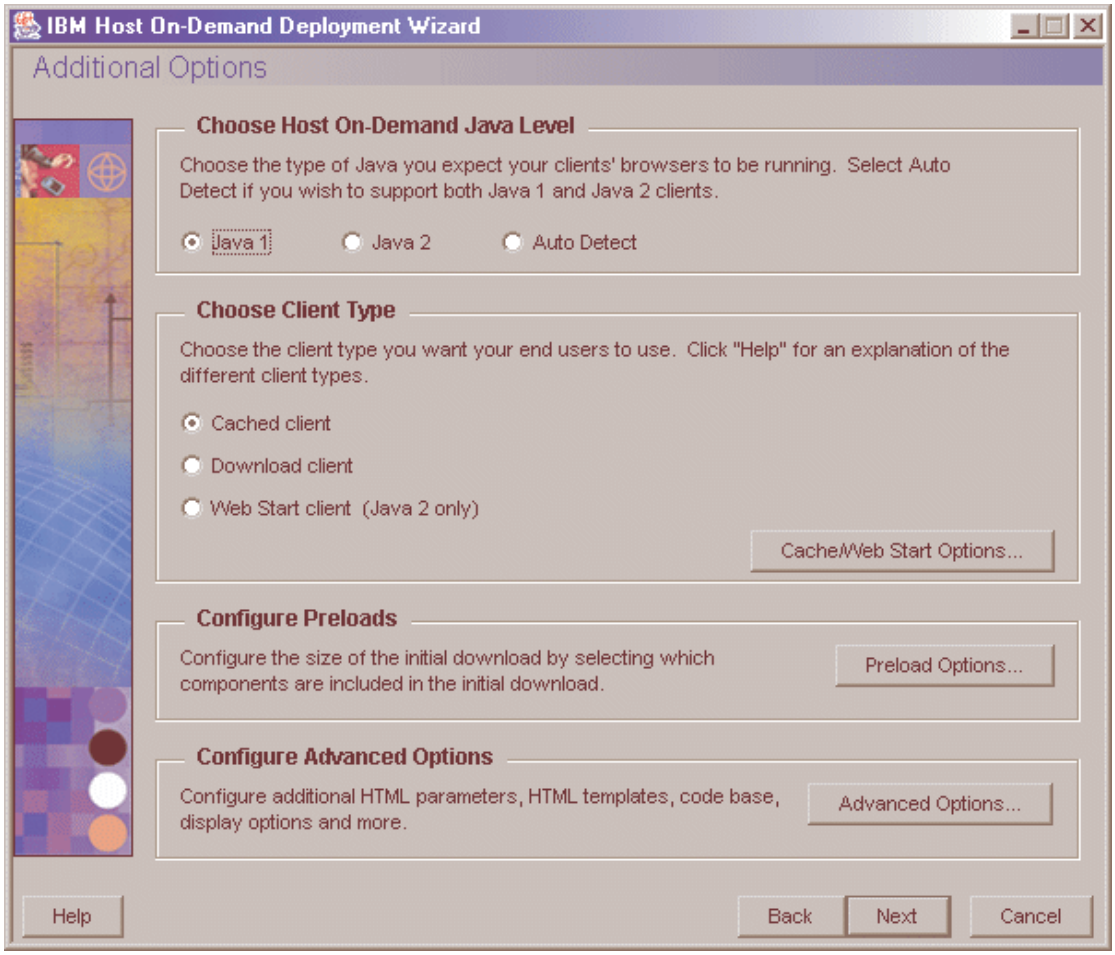


O. Close your session to return to the Host Sessions window.

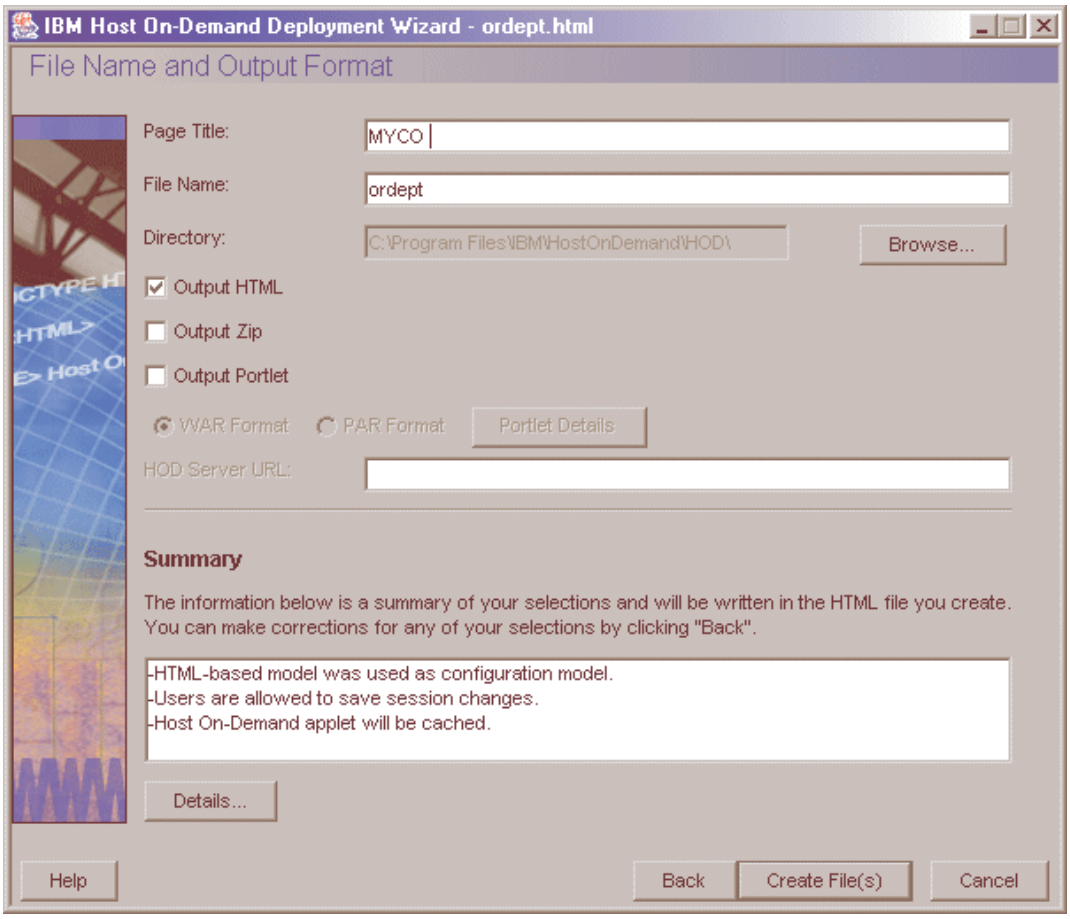
Step 11 of 11: Finish creating your HTML file.

Now that you have configured your Host On-Demand session to use Web Express Logon and have recorded your login macro, you are ready to finish creating your HTML file. At this point, you are still using the Deployment Wizard tool (continued from Step 10).

A. On the Host Sessions window, click Next to open the Additional Options window. Make any changes that you desire and click Next.



B. On the File Name and Output Format window, enter the page title, the file name, and choose the directory where you want to save your file. You should save it to the Host On-Demand server in a directory known to your Web server; usually, this directory is your Host On-Demand server's publish Directory. Click Create File(s) to finish creating your HTML file.



Congratulations! You have now completed Scenario #1: Macro-based automation: Configuring Web Express Logon in a z/OS and DCAS environment. To troubleshoot any problems, refer to Troubleshooting Web Express Logon on page 131.

Scenario #2: Configuring Web Express Logon in a vault-style environment

You are the administrator for a large financial markets/retail banking company. Using Host On-Demand, the company Web-enabled a commercial credit application to provide Internet access to one of the bank's largest customers, a large automobile retailer. This Web-enablement provides better and faster service to the commercial customer who, in turn, provides better and faster service to their customers.

You have just upgraded to Host On-Demand V8 and are planning to implement Web Express Logon so your customers who work at the automobile retailer do not have to log on manually to the commercial credit application, which resides on a z/OS V1R3 host system. In order for this to happen, you must configure your network security application, edit and deploy the CMS provided with Host On-Demand, and create your HCM database. Once that is complete, you use the Deployment Wizard to create your HTML file, configure your 3270 host session, and record your login macro.

Here is a summary of your environment:

- host operating system: z/OS V1R3
- network security application: IBM Tivoli Access Manager for e-business V4.1
- J2EE-compliant Web application server for editing and deploying CMS: IBM WebSphere Application Server V5
- HCM database application: IBM DB2 Universal Database V7

You take the following steps to enable Web Express Logon:

1. Complete the planning worksheet.
2. Configure the network security application (if needed).
3. Create the Host Credential Mapper database.
4. Configure the Credential Mapper Servlet.
5. Deploy the Credential Mapper Servlet.
6. Start creating your HTML file.
7. Configure your Host On-Demand session.
8. Record the Web Express Logon macro.
9. Finish creating your HTML file.

Within this scenario, you will see this FTP icon to highlight points that relate specifically to enabling Web Express Logon for FTP sessions.

Step 1 of 9: Complete the planning worksheet.

The following questions illustrate the type of information you will need before you begin configuring Web Express Logon. The reponses that the administrator gave in this scenario are in the Answers column.

Questions	Answers
What is your host type?	z/OS V1R3
Which network security application do your users go through to access the network?	IBM Tivoli Access Manager for e-business V4.1
Which J2EE-compliant Web application server will you use to edit and deploy your CMS?	IBM WebSphere Application Server V5
Which application will you use to store HCM values?	IBM DB2 Universal Database V7

Step 2 of 9: Configure the network security application (if needed).

If you are using one of the three network security applications that Web Express Logon supports — IBM Tivoli Access Manager, Netegrity Siteminder, or Microsoft Active Directory — you may not need any additional configuration. This document assumes that you already have a network security application in place and have configured any additional steps needed to allow Web Express Logon's Network Security plug-in to acquire the user's network ID. Recall that once Host On-Demand acquires the user's network ID, the HCM database maps it to the user's host ID and password in order to acheive logon automation. If the plug-in cannot acquire this network ID, single sign-on capability will be lost.

In this scenario, the administator has already installed IBM Tivoli Access Manager but needs to perform some additional configuration in order for the Network Security plug-in to acquire the user's network ID successfully. This additional configuration involves WebSEAL, the resource manager component of Tivoli Access Manager that is responsible for inserting the user's network ID into the HTTP header as it passes the request on to the destination host.

In order for WebSEAL to insert the user's network ID into the HTTP header, the administrator must create a junction with a `-c all` option included. To create the junction, he logs in as the sec_master administration user and issues the following `pdadmin> server` task command:

```
pdadmin> server task webseald-cruz create -f -c all -w -t tcp -h
dtawg.raleigh.ibm.com -p 80 /junction
```

where *`webseald-cruz`* is the name of the Tivoli Access Manager server host name, *`dtawg.raleigh.ibm.com`* is the fully qualified domain name of the back-end server, *`80`* is the port number (this is the default), and *`junction`* is the name of the junction point.

For more information about Tivoli Access Manager, WebSEAL, and creating WebSEAL junctions, refer to the following Web site:

http://publib.boulder.ibm.com/tividd/td/IBMAccessManagerfore-business4.1.html

Step 3 of 9: Create the Host Credential Mapper database

The Host Credential Mapper (HCM) database is one of the key players in the Web Express Logon process because it maps users' network IDs to their host IDs. Since the Vault parameters supplied with Web Express Logon are designed to work with a JDBC database, the administrator in this scenario uses IBM DB2 to create the HCM database. Using this type of network-accessible database provides a flexible and secure means of associating users' network IDs with their host IDs.

Using DB2, the administrator creates a table with column names that correspond to the Vault parameters that he will add to the servlet configuration file in the next step. The following five column names are in all upper case and must exactly match the parameter values he will specify in the servlet configuration file:

- NETWORKID: This column contains the network IDs of the users. A user's network ID is the credential that uniquely identifies the user to the network security application (in this case, Tivoli Access Manager).
- HOSTADDRESS: The column contains the destination host address. This address can either be the host's IP address or the fully qualified URL, for example, amin.raleigh.ibm.com.
- APPLICATIONID: This column contains the application IDs of the users. Application IDs are used to map users' host IDs and to retrieve passwords from the vault-style database.

The APPLICATIONID column is not required for FTP sessions.

- HOSTID: This column contains the users' host IDs. A host ID is the credential used to uniquely identify the user to the host being accessed.
- PASSWORD: This column contains the users' host passwords. Similar to a DCAS passticket, this password is used to map the users' network IDs to the host IDs; however, it requires an additional parameter in the servlet configuration file.

For more information about IBM DB2, refer to the following Web site: http://www.ibm.com/software/data/db2/library/**.**

Step 4 of 9: Configure the Credential Mapper Servlet.

In this step, you will configure the Credential Mapper Servlet (CMS). The CMS is the core of the credential-mapping framework. It is supplied with Host On-Demand and must be deployed to a J2EE-compliant Web application server. At a high level, the CMS is responsible for the following tasks: (1) determine the client's identity (called a network ID), (2) map the user's network ID to the host ID, and (3) return the host credentials to the client as an XML document.

Host On-Demand provides three CMS WAR files, one for each of the following network security applications:

- IBM Tivoli Access Manager for e-business V4.1
- Netegrity Siteminder V5.5
- Microsoft Active Directory (Windows Domain)

If you have a different network security application, you will need to customize your own version of the CMS. For more information about how to do this, refer to Customizing Web Express Logon on page 124.

In addition to several CLASS files, the WAR files contains the following four files:

- web.xml
- DCAS.xml
- Vault.xml
- was.policy

The web.xml file is the servlet configuration file that you will edit in this step. The other two XML files (DCAS.xml and Vault.xml) are sample files that we have provided to help you better understand DCAS and Vault parameters and their values. We also recommend that you use these files as a reference when you edit the web.xml file. Finally, the was.policy file is for IBM WebSphere Application Server only. It contains the required permissions for the CMS when Java 2 security is enabled. For more information, refer to Troubleshooting Web Express Logon on page 131.

In this scenario, the administrator uses WebSphere Application Server to configure the CMS. He takes the following steps:

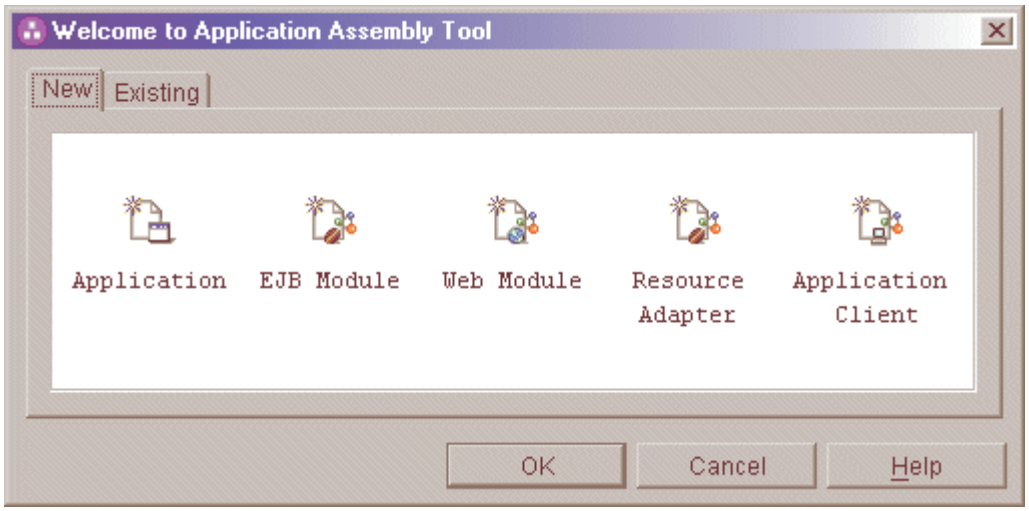
A. Locate the WAR files on the Host On-Demand CD.
B. Edit the CMS-related parameters.
C. Add optional CMS-related debugging parameters.
D. Add the required Vault parameters.
E. Add optional Vault parameters (if desired)
F. Save the WAR file.

****

A. Locate the WAR files on the Host On-Demand CD. With the Host On-Demand V8 CD loaded in the CD drive of your machine, take the following steps to locate the WAR files on the CD.

i. Click Start > Programs > IBM WebSphere > Application Server v5.0 > Application Assembly Tool.

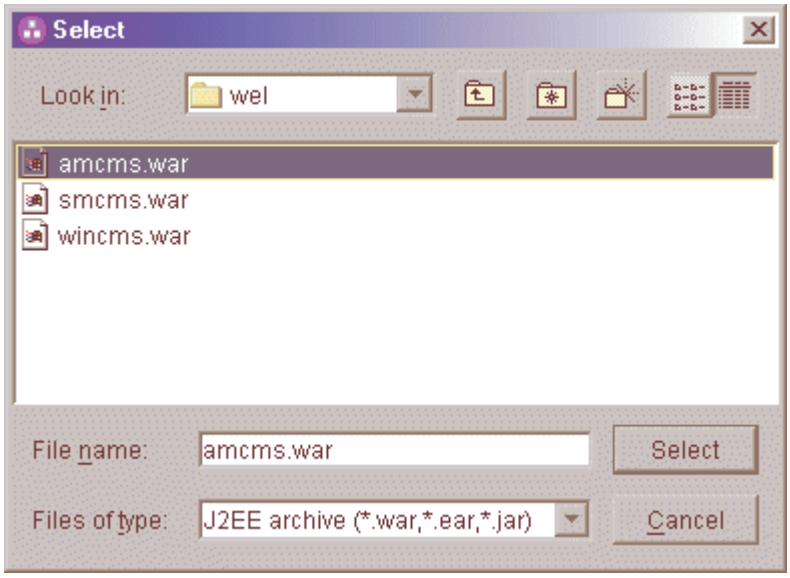
ii. On the Welcome to the Application Assembly Tool window, click the Existing tab and browse to the apps\wel directory on the Host On-Demand CD.



iii. In the apps\wel directory, you will see three WAR files, one for each of the following network security applications:

Network security application	Corresponding WAR file
IBM Tivoli Access Manager for e-business V4.1	amcms.war
Netegrity Siteminder V5.5	smcms.war
Microsoft Active Directory (Windows Domain)	wincms.war

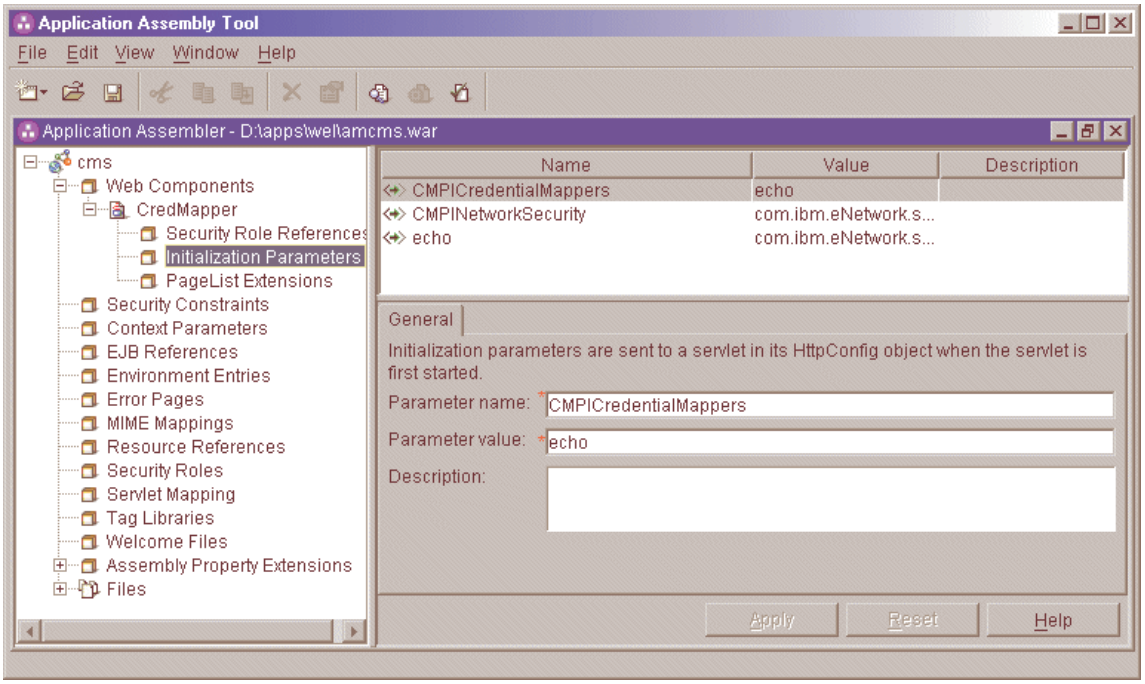
Highlight the WAR that represents your network security application and press Select. Then select OK.



If you have a different network security application, you will need to customize your own version of the CMS. For more information about how to do this, refer Customizing Web Express Logon on page 124.

B. Edit the CMS-related INIT parameters: In this step, you will edit two of the three INIT parameters in the web.xml file. You will not edit the CMPINetworkSecurity parameter name or value.

i. In the left panel of the window, expand Web Components > CredMapper and click Initialization Parameters. The three default INIT parameters specifically coded to adapt the CMS to your environment appear in the top window.



(1) - (3) describe these default parameters in more detail:

(1) Host Credential Mapper (HCM) plug-in: The name of the parameter is CMPICredentialMappers, and the parameter value is a compound value that contains the list of all available HCM plug-ins, for example, CMPIDCASPlugin and CMPIVaultPlugin. Currently, the value is echo, but you will eventually replace this with the name of your HCM plug-in.

Code example:

```
<init-param>
   <param-name>CMPICredentialMappers</param-name>
   <param-value>echo</param-value>
</init-param>
```

(2) Network Security plug-in: The name of the parameter is CMPINetworkSecurity, and the parameter value is the full path name of the class that handles the CMS interface into the network security application, which is Tivoli Access Manager in this scenario.

Code example:

```
<init-param>
   <param-name>CMPINetworkSecurity</param-name>

<param-value>com.ibm.eNetwork.security.sso.cms.CMNPIAccessManager</pa
ram-value> </init-param>
```

The Network Security plug-in does not apply to Microsoft Active Directory XML file (wincms.xml) since the Windows login ID is used as the network ID.

(3) echo plug-in: The name of this INIT parameter (echo) is the same as the value for the HCM plug-in. In a future step, you will replace echo with the name of your HCM.

Host On-Demand provides this optional echo plug-in in case you want to confirm that you are able to deploy the CMS correctly before you begin editing the web.xml file. For example, after you deploy your CMS to a Web server, you can test it by entering the following syntax in a PC's browser address bar: https://web_application_server_name/context_root/CredMapper, where web_application_server_name is the name of the Web application server, context_root is the name of the context root that you specified when deploying the CMS, and CredMapper is the name of the CMS itself.

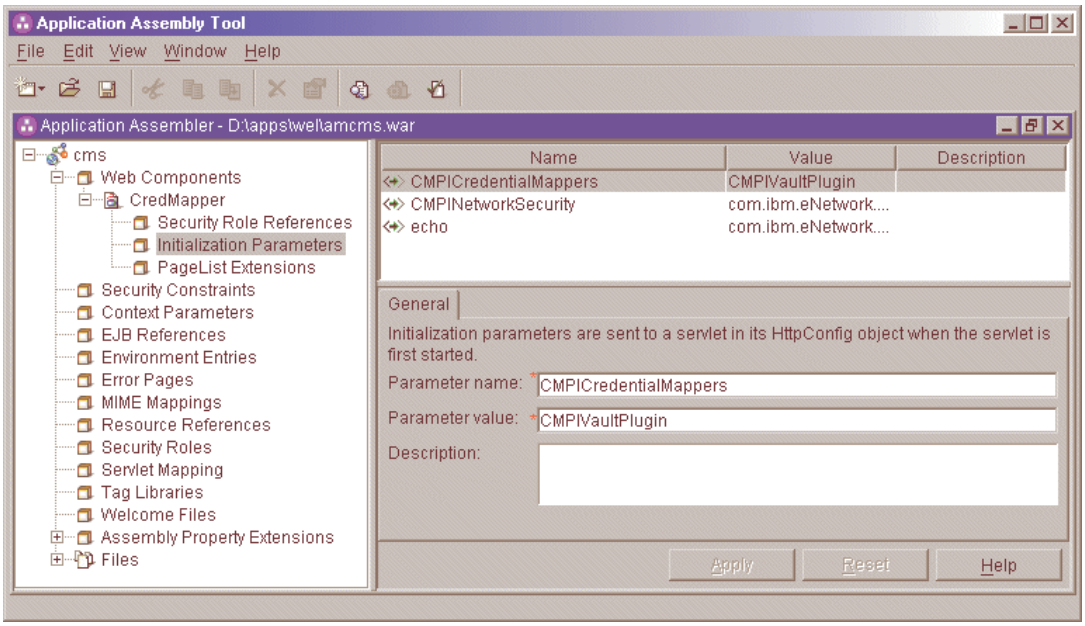
Some Web application server products allow you to deploy the servlet first and then edit the XML file. Other products, such as WebSphere Application Server V5, work best when you deploy the servlet after you edit the XML code. Refer to your product's documentation for details.

Code example:

```
<init-param>
    <param-name>echo</param-name>
          <param-value>com.ibm.eNetwork.security.sso.cms.CMPINetEcho,
          AuthType_All,*</param-value>
</init-param>
```

ii. Highlight the CMPICredentialMappers parameter in the top panel of the window. In the Parameter value field below, change the name of its current value (echo) to the name of your HCM plug-in. In this scenario, the administrator specifies CMPIVaultPlugin as the parameter value because he is using a vault-style database as his HCM plug-in.

Optionally provide a description and click Apply to replace the value in the top Window.



Code example:

```
<init-param>
   <param-name>CMPICredentialMappers</param-name>
   <param-value>CMPIVaultPlugin</param-value>
</init-param>
```

iii. Now highlight the echo parameter. In the Parameter name field, replace the current parameter name (echo) with the name of the parameter value that you specified for the HCM plug-in. In this scenario, the administrator changed the parameter name to CMPIVaultPlugin.

Now, replace the parameter value with a compound value that contains the full class path name of the implementing class, the authentication type to be used by the HCM plug-in, and the host mask. Separate these values with commas. In this scenario, the administrator added com.ibm.eNetwork.security.sso.cms.CMPIVault for the full class path name, AuthType_All for the authentication type, and * for the host mask.

The authentication type for FTP sessions should be either AuthType_FTPPassword or AuthType_All.

Full class path name
The CMS uses the value of the full class path name to create a class object of the specified type. That object is then used to handle CMS or HCM plug-in requests. The specified class file must be in the ...\WEB-INF\classes subdirectory in a loose file (not as a JAR file). From this location, the CMS will be able to access and use it whenever the need arises.

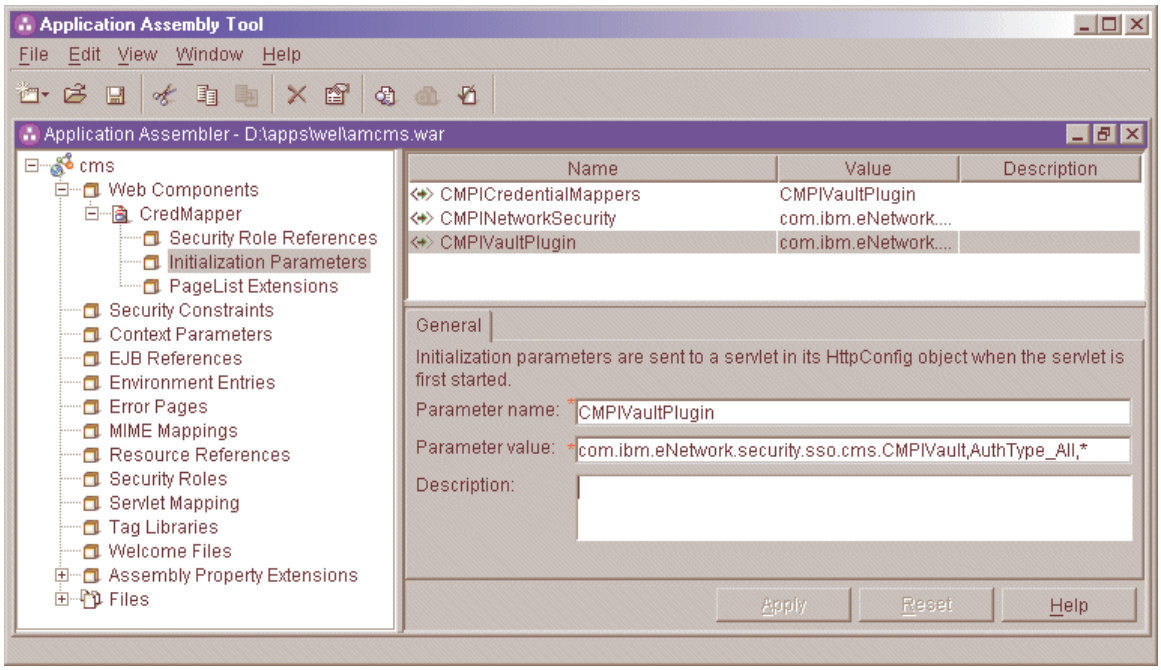
Authentication type
This parameter value is used to identify the type of authentication that the requestor needs. Once you specify the desired authentication type, the CMS can better identify which HCM plug-in to select to handle the request. You can pair multiple authentication types together to give HCM plug-ins the freedom to support multiple authentication types. Use the vertical bar character to join multiple authentication types. The five identified authentication types and descriptions are listed in the following table:

Authentication type	**Description**
AuthType_3270Host	Identifies the credentials to be used with a 3270 emulation
AuthType_5250Host	Identifies the credentials to be used with 5250 emulation
AuthType_VTHost	Identifies the credentials to be used with VT emulation
AuthType_FTPPassword	Credentials used to access an FTP host
AuthType_ConfigServer	Credentials identified by the token used to identify the user to the Host On-Demand configuration server (if you are using the Configuration server-based model)
AuthType_All	Identifies the credentials to be used with all authentication types

Host mask
The host mask is a secondary selection criteria used by the CMS to identify the most appropriate HCM plug-in. This value can contain one or more host addresses. Use the vertical bar character to join multiple addresses. Use the asterisks character to wildcard a host address. The wildcard character may start, end, or start and end a host address. The following table lists valid wild-carded addresses:

Host mask	**Value matched**
*.raleigh.ibm.com	Matches all addresses that end with .raleigh.ibm.com
ralvm*	Matches all addresses that start with ralvm
*	Matches all
xyz	Matches any host address that contains xyz

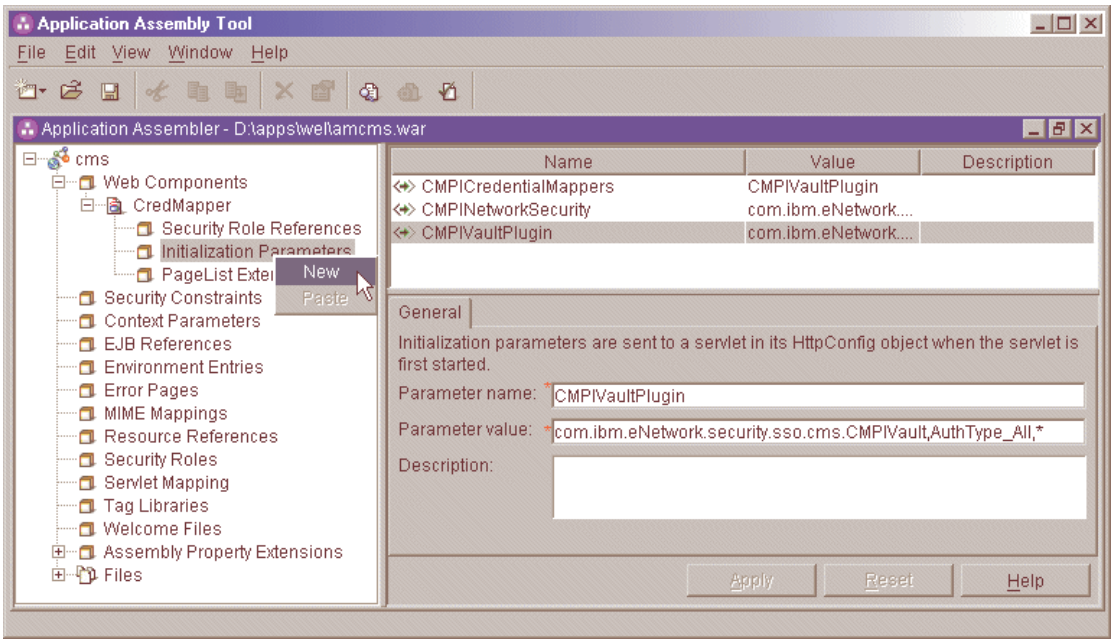
Optionally provide a description and click Apply to replace the value in the top window.



Code example:

```
<init-param>
   <param-name>CMPIVaultPlugin</param-name>
<param-value>com.ibm.eNetwork.security.sso.cms.CMPIVault,
   AuthType_ALL,*</param-value>
</init-param>
```

C. Add optional CMS-related debugging parameters: To add new parameters, right-click Initialization Parameters in the left pane of the Application Assembly Tool window and select New.



Using the New Initialization Parameter window, add the following two optional debugging parameters to help you troubleshoot:

CMPI_TRACE_LOG_FILE

This parameter specifies the name of the log file. The value should be the full path to the log file, for example C:\Program Files\IBM\HostOnDemand\HOD\HODWEL.log on a Windows platform.

Code example:

```
<init-param>
   <param-name>CMPI_TRACE_LOG_FILE</param-name>
   <param-value>C:\Program
Files\IBM\HostOnDemand\HOD\HODWEL.log</param-value>
</init-param>
```

CMPI_CMS_TRACE_LEVEL

This parameter specifies the trace level for the CMS. The trace messages are logged to the log file specified by CMPI_TRACE_LOG_FILE parameter. Depending on your Web application server, they may or may not be logged to the console. Trace level values include the following:
0 = None: No tracing. This is the default.
1 = Minimum: Trace APIs and parameters, return values, and errors.
2 = Normal: Trace Minimum plus internal APIs and parameters and informational messages.
3 = Maximum: Trace Normal plus Java exceptions.

Code example:

```
<init-param>
   <param-name>CMPI_CMS_TRACE_LEVEL</param-name>
   <param-value>3</param-value>
</init-param>
```

D. Add the required Vault parameters for the CMPIVaultPlugin: Adding the required Vault parameters allows the HCM to map the user's network ID to his host ID and receive the needed password from the vault-style database. The following Vault parameters are required in order for Web Express Logon to function properly. This section is divided into two subsections, I and ii.

i. The following parameters contain all the relevant information needed to connect to your HCM, which in this case is a JDBC database table created with IBM DB2. You can either configure access to an existing database or to a newly created one. The level of security for the database depends on the database vendor. See the documentation for details.

CMPI_VAULT_DB_ADDRESS

This is a URL string that provides the address of the database. An example of this string is jdbc:db2://dtagw.raleigh.ibm.com:6789/HODSSO.

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_ADDRESS</param-name>

<param-value>jdbc:db2://dtagw.raleigh.ibm.com:6789/HODSSO</param-value>
</init-param>
```

CMPI_VAULT_DB_NET_DRIVER

This string contains the name of the class that acts as the network database driver. An example of this string is COM.ibm.db2.jdbc.net.DB2Driver. The location of this class is assumed to be in the existing class path.

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_NET_DRIVER</param-name>
   <param-value>COM.ibm.db2.jdbc.net.DB2Driver</param-value>
</init-param>
```

CMPI_VAULT_DB_USERID

This is the ID of the user account to use when accessing the database. In this case, the user ID is admin.

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_USERID</param-name>
   <param-value>admin</param-value>
</init-param>
```

CMPI_VAULT_DB_PASSWORD

This is the password of the user account to use when accessing the database. We strongly recommend that you encrypt this parameter using the password encryption tool provided with Host On-Demand. The tool encrypts the password and then decrypts it so the HCM plug-in can use it. To learn more about how to use this tool, refer to Password Encryption Tool on page 140.

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_PASSWORD</param-name>
   <param-value>tuBu9v8lHiJi1jt08UgHzA==</param-value>
```

```
</init-param>
```

CMPI_VAULT_DB_TABLE
This identifies the table to use for the needed query. In this case, the table is called HACP.

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_TABLE</param-name>
   <param-value>HACP</param-value>
</init-param>
```

ii. The following parameters correspond directly to the column names that you added to your HCM database table in Step 3: Create the Host Credential Mapper database. Recall that you added the following five column names, all in uppercase: NETWORKID, HOSTADDRESS, APPLICATIONID, HOSTID, and PASSWORD.

The APPLICATIONID column is not required for FTP sessions.

Based on the information provided by the first three of these parameters (network ID, host address, and the host application ID), you can make a SQL query of the database to get the host ID. The result of the query is entered in the host ID (HOSTID) column. Assuming that the query is successful, a call is made to the vault-style database to request the password.

CMPI_VAULT_DB_NETID_COL_NAME
This entry identifies the name of the column that contains the network ID value (NETWORKID).

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_NETID_COL_NAME</param-name>
   <param-value>NETWORKID</param-value>
</init-param>
```

CMPI_VAULT_DB_HOSTADDR_COL_NAME
This entry identifies the name of the column that contains the host address value (HOSTADDRESS).

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_HOSTADDR_COL_NAME</param-name>
   <param-value>HOSTADDRESS</param-value>
</init-param>
```

CMPI_VAULT_DB_HOSTAPP_COL_NAME
This entry identifies the name of the column that contains the host application value (APPLICATIONID).

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_HOSTAPP_COL_NAME</param-name>
```

```
   <param-value>APPLICATIONID</param-value>
</init-param>
```

CMPI_VAULT_DB_HOSTID_COL_NAME
This entry identifies the name of the column that contains the host ID value (HOSTID).

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_HOSTID_COL_NAME</param-name>
   <param-value>HOSTID</param-value>
</init-param>
```

CMPI_VAULT_DB_HOSTPW_COL_NAME
This entry identifies the name of the column that contains the host password value (PASSWORD).

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_HOSTPW_COL_NAME</param-name>
   <param-value>PASSWORD</param-value>
</init-param>
```

E. Add any optional Vault parameters: Unlike the previous set of Vault parameters, the following parameters are optional. Which of these parameters you add to the web.xml file depends on your environment and your objectives as an administrator:

CMPI_VAULT_TRACE_LEVEL
This parameter specifies the trace level for the Vault plug-in. The trace messages are logged to the log file specified by CMPI_TRACE_LOG_FILE parameter. Depending on your Web application server, they may or may not be logged to the console. Trace level values include the following:
0 = None: No tracing. This is the default.
1 = Minimum: Trace APIs and parameters, return values, and errors.
2 = Normal: Trace Minimum plus internal APIs and parameters and informational messages.
3 = Maximum: Trace Normal plus Java exceptions.

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_TRACE_LEVEL</param-name>
   <param-value>3</param-value>
</init-param>
```

CMPI_VAULT_DB_PRESERVE_WHITESPACE

This parameter indicates whether to trim white spaces from the credential request parameters or not. If true, the white spaces are not trimmed. The default is false.

Code example:

```
<init-param>
   <param-name>CMPI_VAULT_DB_PRESERVE_WHITESPACE</param-name>
   <param-value>false</param-value>
</init-param>
```

F. Save the WAR file: On the Application Assembly Tool window, click File > Save As to save your WAR file to your preferred location. If it saves successfully, this window will appear as a confirmation:

Step 5 of 9: Deploy the Credential Mapper Servlet.

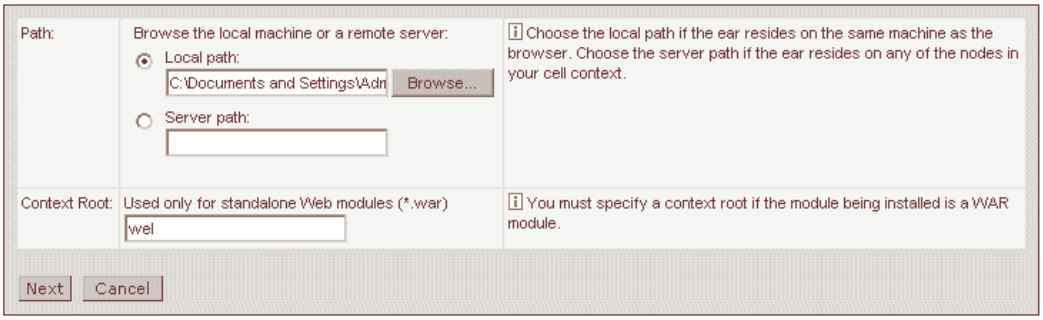
The way in which you deploy the CMS depends on your Web application server. In this scenario, the administrator uses WebSphere Application Server V5 on a Windows platform to deploy the CMS. He takes the following steps:

A. Click Start > Programs > IBM WebSphere > Application Server v5.0 > Administrative Console to open the Administrative Console.

B. In the left navigation field, click Applications > Install New Application.

C. Select Local path and browse to the amcms.war file.

D. Specify the content root in the Context Root field. The context root is combined with the defined servlet name (CredMapper) to compose the full URL that users type to access the servlet. For example, if the context root is /wel , then the URL is https://host:port/wel/CredMapper, where host is the name of the host, port is the port number, and CredMapper is the name of the CMS. Click Next.



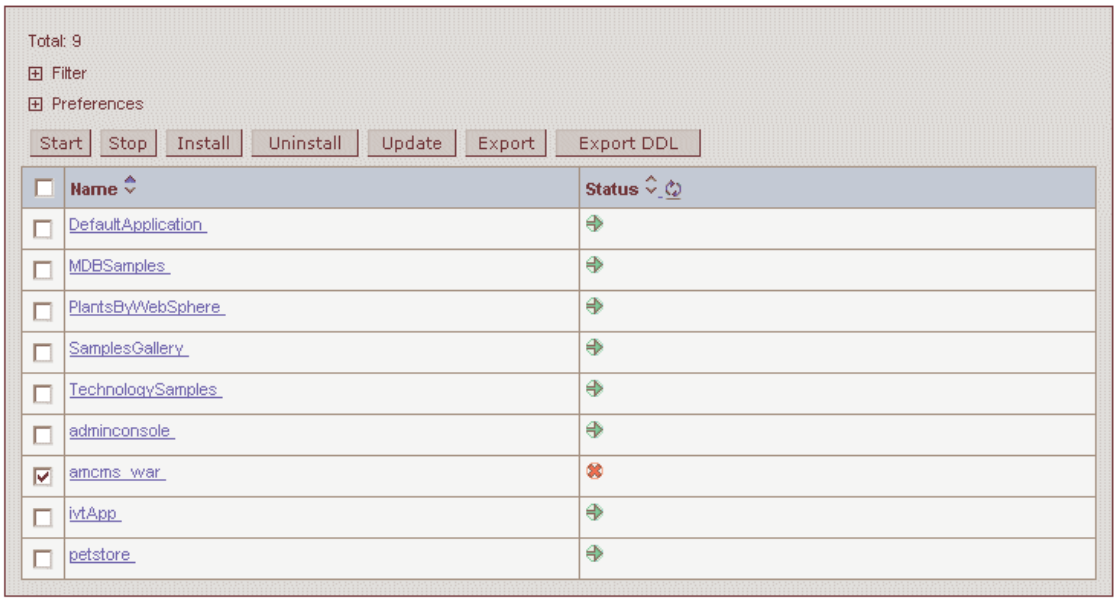
E. Check the Generate Default Bindings box. By choosing this option, you can jump directly to the Summary step and deploy the WAR file. Click Next.

F. Scroll down and click the Step 4 Summary link.

G. Scroll down and click Finish.

H. In the left navigation field, click Applications > Enterprise Applications.

I. Scroll through the applications and check the checkbox beside amcms_war, which is your application name. Notice that the status is Stop until you start it in the next step.



J. Save the application and click Start at the top to start the application. Once the application starts, the status will change from the Stop icon,

to the Start icon

.

K.Test to make sure that the WAR file installed correctly by pointing a browser to http://server_name:9080/wel/CredMapper, where server_name is the name of the host server. Port 9080 is the default WebSphere port. If you deployed the servlet correctly, your browser request will return XML code. If this fails, try stopping and restarting the application again.

Step 6 of 9: Begin creating your HTML file.

The Host On-Demand Deployment Wizard allows you to create an HTML file that is used to launch Host On-Demand sessions. Within the Deployment Wizard, you can add, delete, configure, and start sessions. It guides you configuration choices and provides comprehensive help for the features. When you have finished selecting features, it creates the HTML and supporting files for you.

In this scenario, the administrator performs Steps 6 - 9 all within the Deployment Wizard in one sitting. However, you may decide to create your HTML file first and then configure your session and create your macro later. Refer to the supplemental notes at the beginning of Steps 7 and 8 for more information about how to do this.

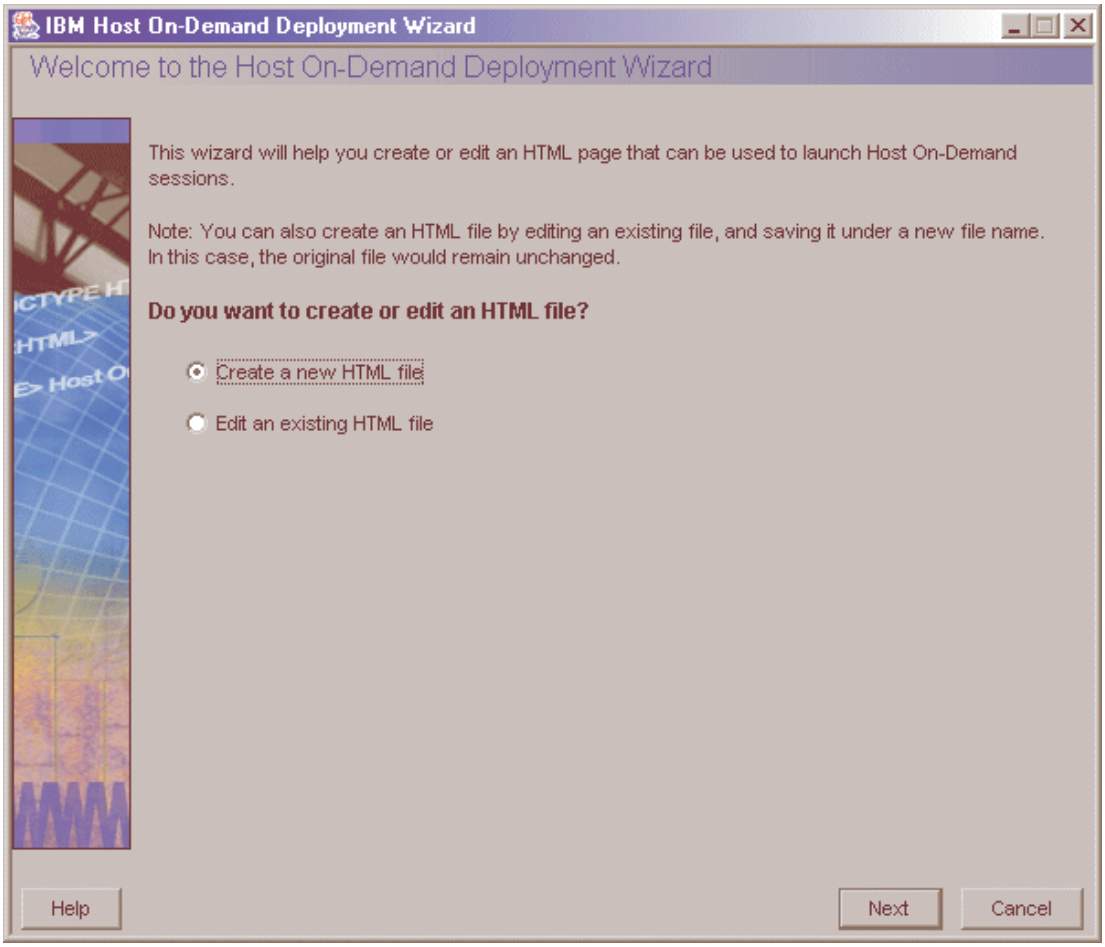
To begin creating your HTML file, take the following steps:

A. Open the Deployment Wizard:

- If you automatically installed the Deployment Wizard as part of the Windows Host On-Demand server, click Start > Programs > IBM WebSphere Host On-Demand > Administration > Deployment Wizard.
- If you installed the Deployment Wizard from the Host On-Demand CD separately, click Start > Programs > IBM WebSphere Host On-Demand Deployment Wizard > Deployment Wizard.

Click Start > Programs > IBM WebSphere Host On-Demand > Administration > Deployment Wizard.

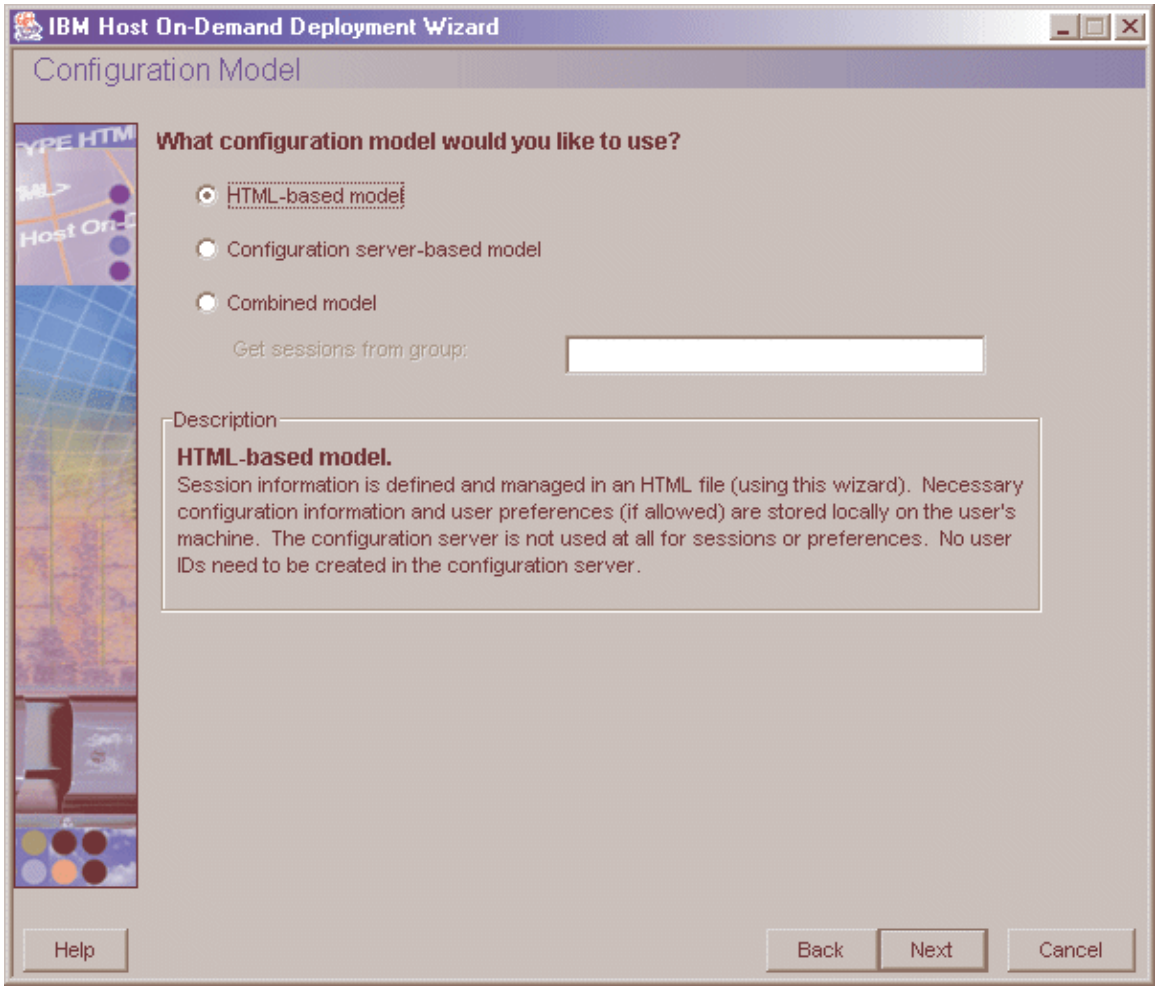
B. Select either to create a new HTML file or edit an existing file. Click Next.



C. Select from the following three configuration models and click Next:

- HTML-based model
- Configuration server-based model *(*if you choose this configuration model, refer to 121.)
- Combined model

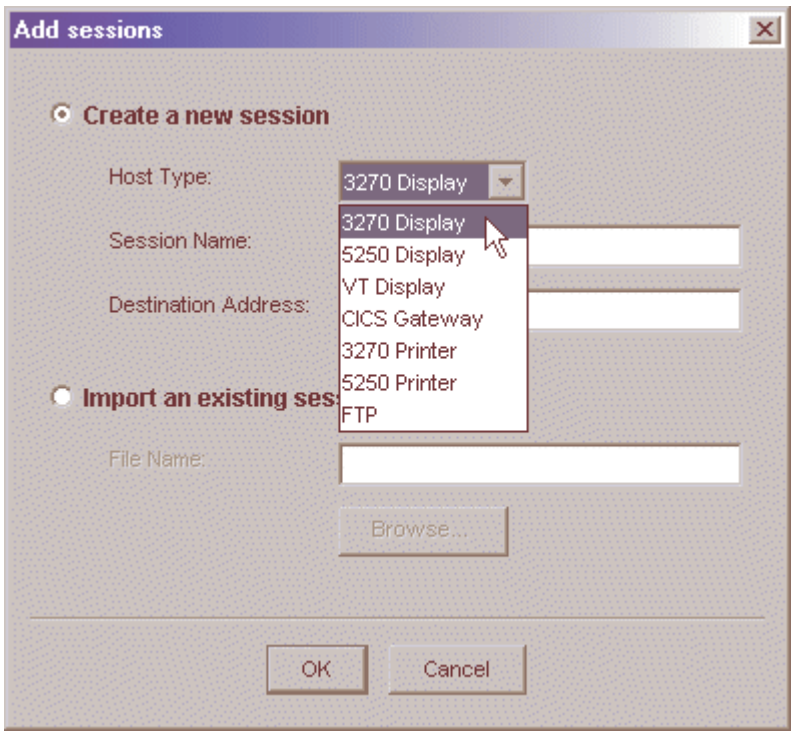
The administrator in this scenario selects the HTML based-model.



D. On the Host Sessions window, click New/Import to open the Add sessions window. This window allows you to either create a new session (default) or import an existing session.

To create a new session, select a host type, enter a session name, and a destination address. In this scenario, the administrator selects 3270 Display.

For FTP sessions, select FTP as the host type.



Click OK to return to the Host Sessions window.

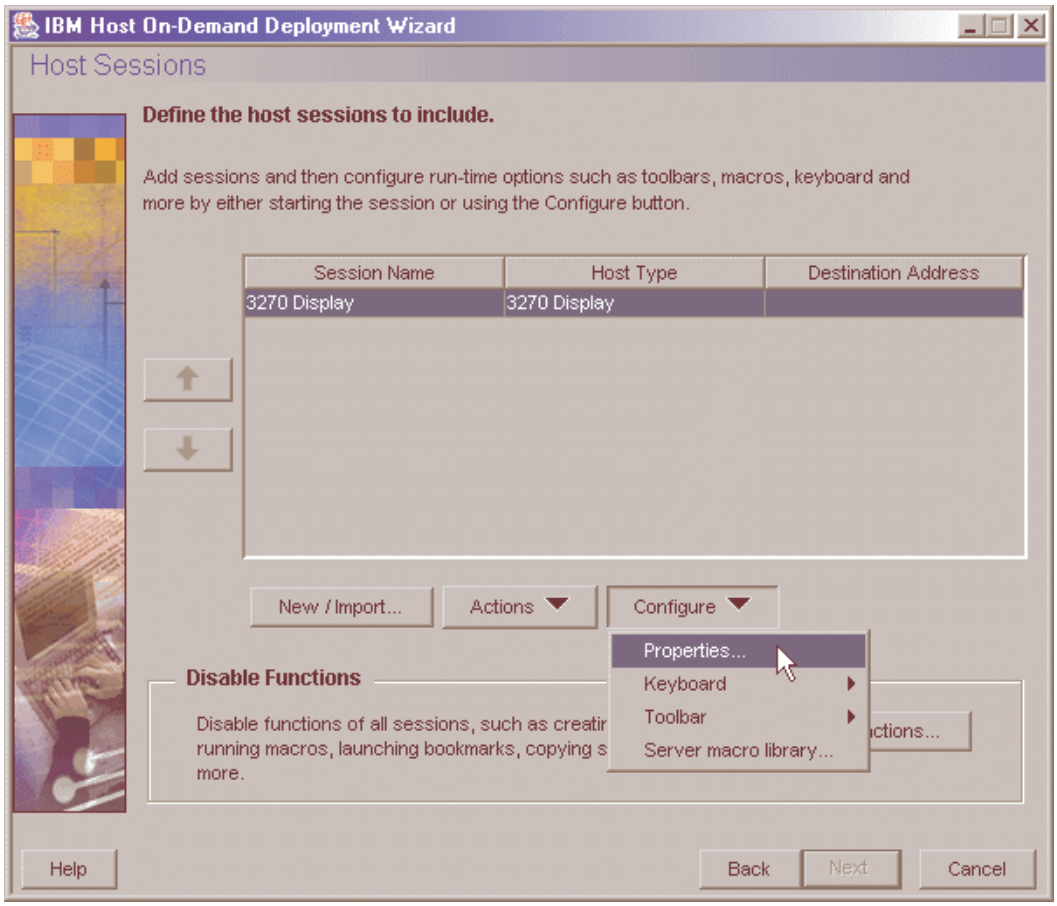
Step 7 of 9: Configure your Host On-Demand session.

In this step, you will configure your Host On-Demand session to use Web Express Logon. At this point, you are still using the Deployment Wizard tool (continued from Step 6).

If you have already created your HTML file and now wish to configure it to use Web Express Logon, open the Host On-Demand desktop, right-click the session icon, and select Properties. Skip to Step B.

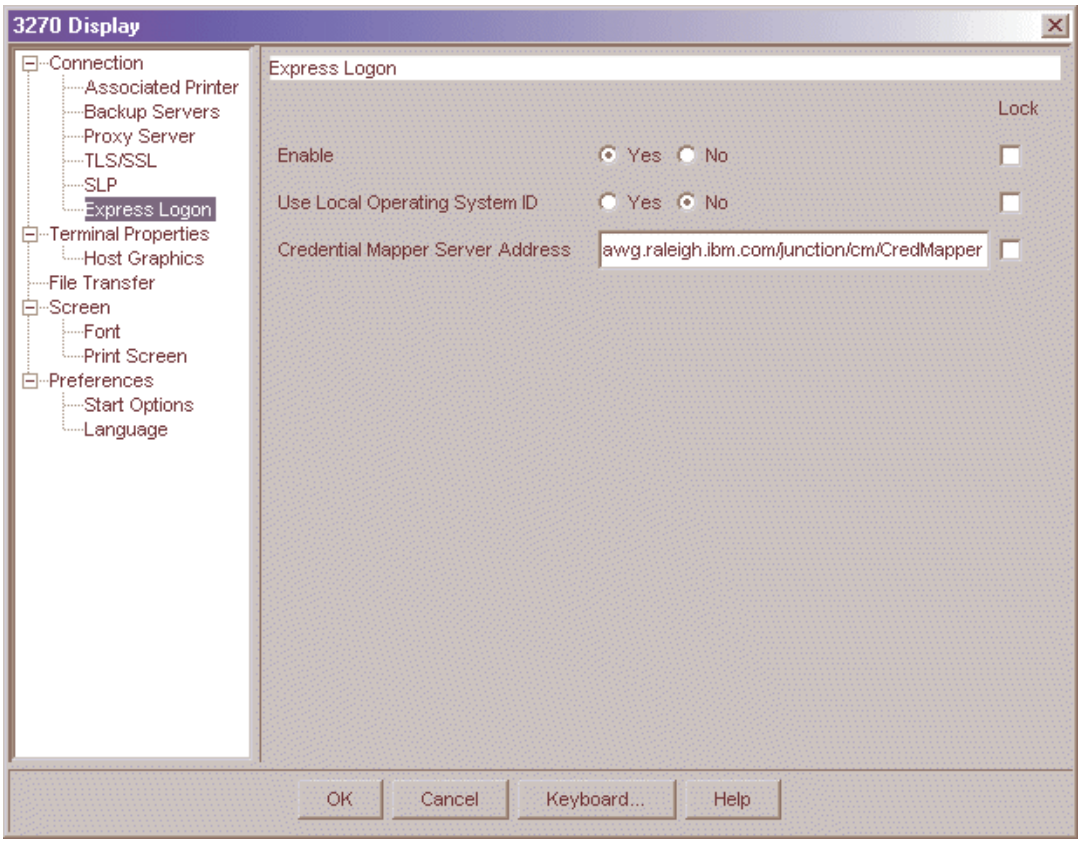
A. On the Host Sessions window, click Configure > Properties to configure your session to use Web Express Logon.

If you are configuring an FTP session, your Session Name will be FTP, and your Host Type will be FTP/sftp.



B. Under the Connection option on the left side of the window, click Express Logon. Select Yes to enable Express Logon and chose whether or not you want Host On-Demand to use the user's local operating system ID for authentication. Next, type the full URL of the credential mapper server, for example, https://*server_name*/*junction*/*cm*/*CredMapper*, where

- *server_name* is the name of the authentication server
- *junction* is the name of the junction point (optional)
- *cm* is the credential mapper servlet space
- *CredMapper* is the servlet name



Be sure that the servlet name matches the name in your XML file. For example, if you specify the servlet name in your host session as CredMapper (recommended), the code in your XML should look like the following:

```
<servlet>
   <servlet-name>CredMapper</servlet-name>
   <display-name>CredMapper</display-name>

<servlet-class>com.ibm.eNetwork.security.sso.cms.CredMapper</servlet-cla
ss>
```

The servlet that resides at this URL processes the HTTPS request from the user, performs a lookup, and returns the user's credentials. The Host On-Demand client uses the obtained credentials to automate the login process.

When configuring properties for FTP sessions, there is a Logon option in the left panel of the window. On this panel, be sure that you leave the User ID and Password fields blank if you are enabling Web Express Logon. If you add a user ID and/or password, Host On-Demand will ignore the settings on the Express Logon panel.

Click OK to return to Host Sessions window.

Step 8 of 9: Record the Web Express Logon macro.

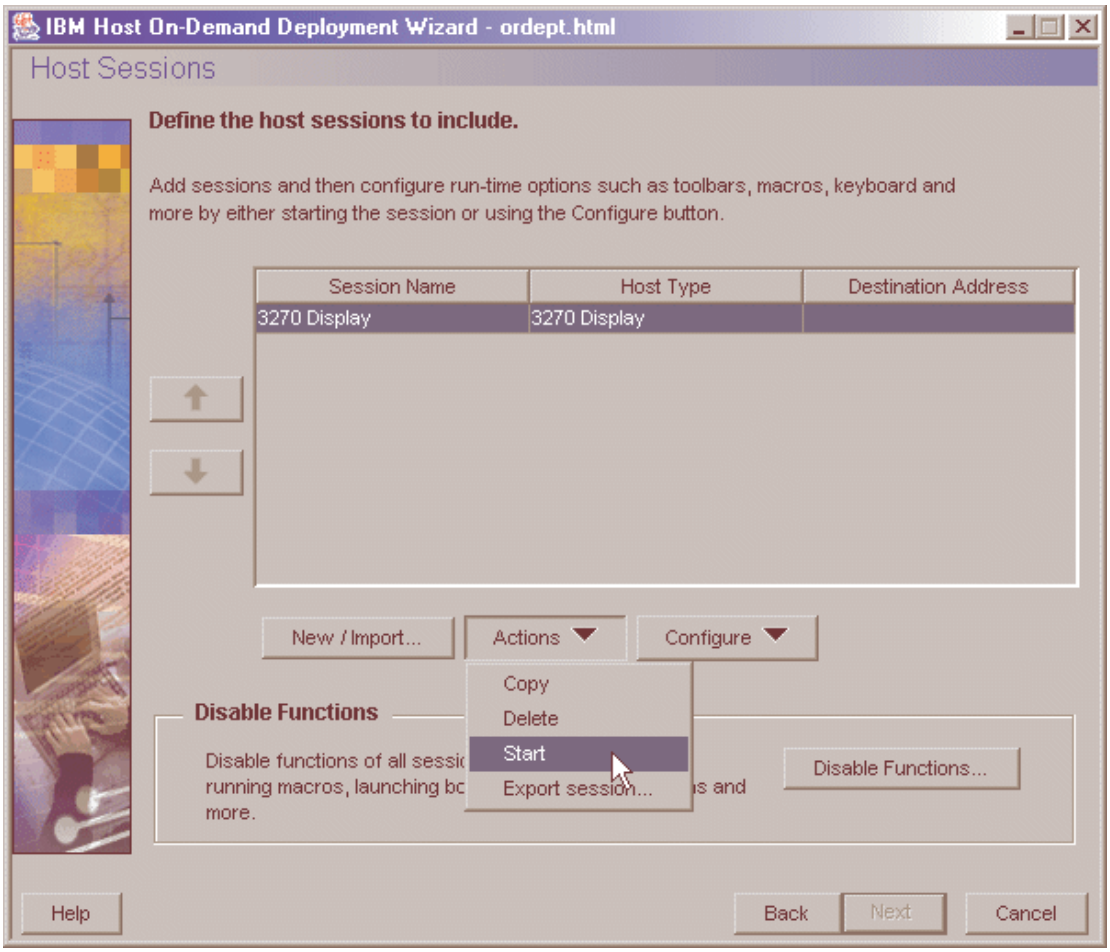
In this step, you will create the component that ties the whole logon automation process together — the login macro. You record this during an active session. At this point, you are still using the Deployment Wizard tool (continued from Step 7).

Step 8 of 9: Record the Web Express Logon macro is not required for FTP sessions.

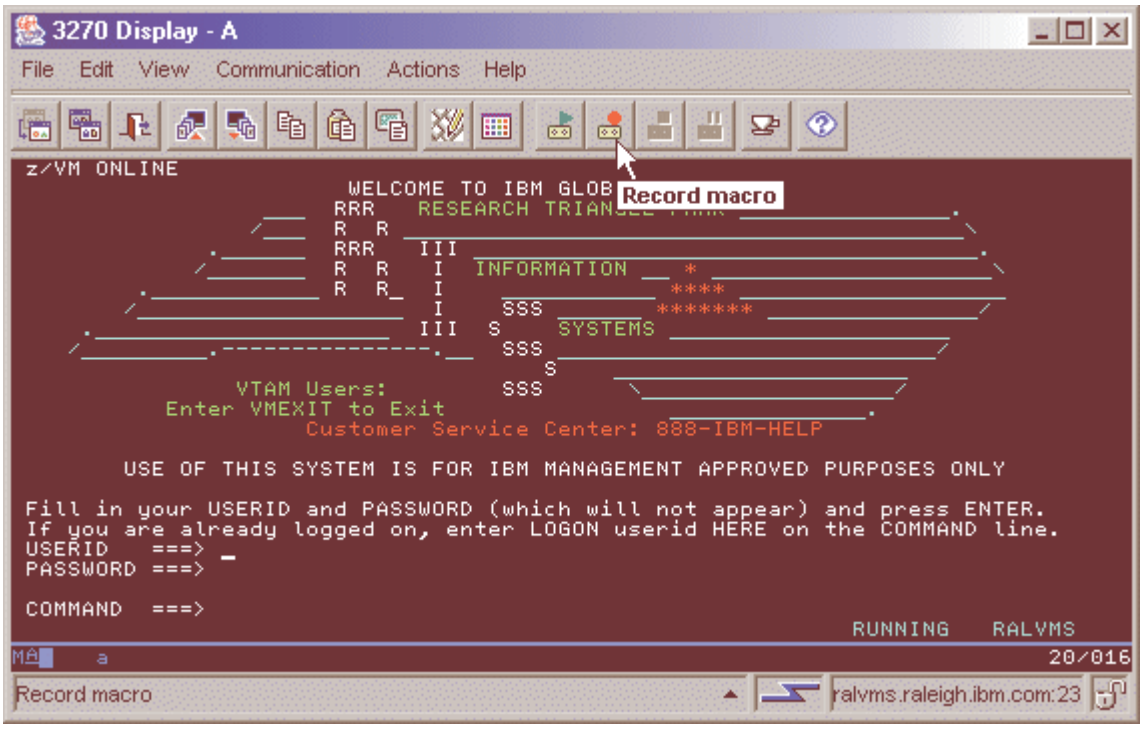
If you have already created your HTML file and now wish to record the login macro, open the Host On-Demand desktop, right-click the session icon, and select Start. Skip to Step B.

To record the macro, take the following steps:

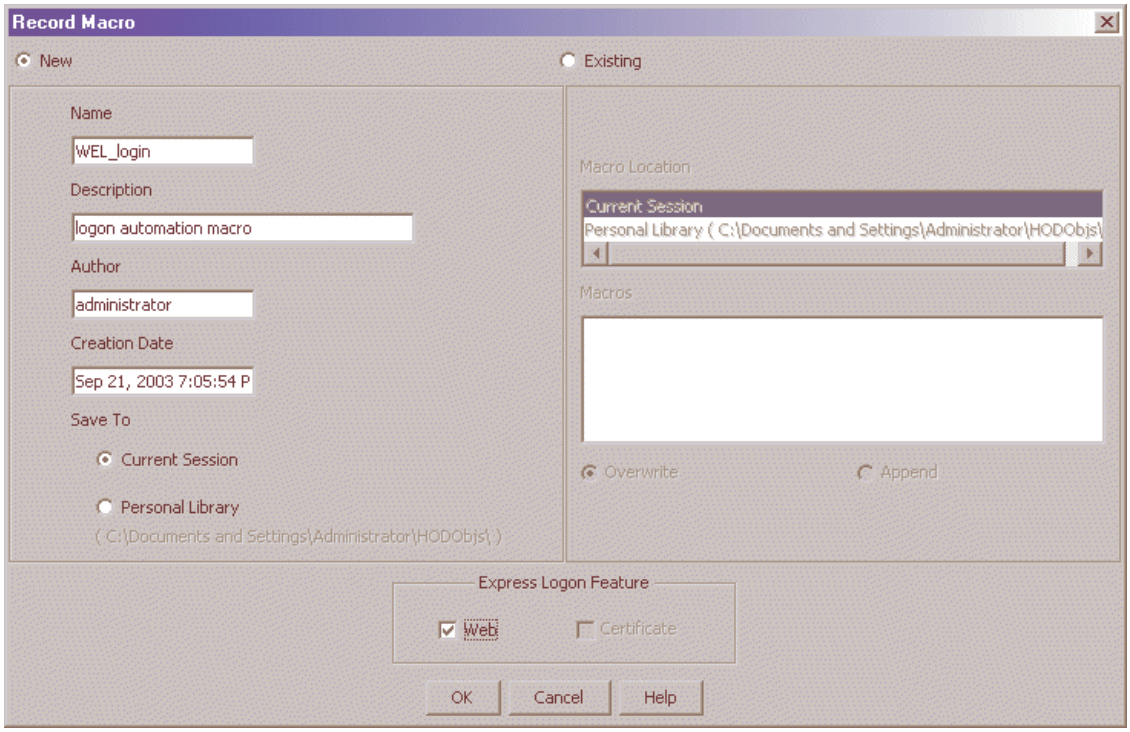
A. On the Host Sessions window, click Actions > Start to start your Session.



B. Click the Record macro button on the toolbar of your active session.



C. On the Record Macro window, select New and fill in the Name and Description (optional) fields. Check Web under Express Logon Feature at the bottom of the window. Click OK.



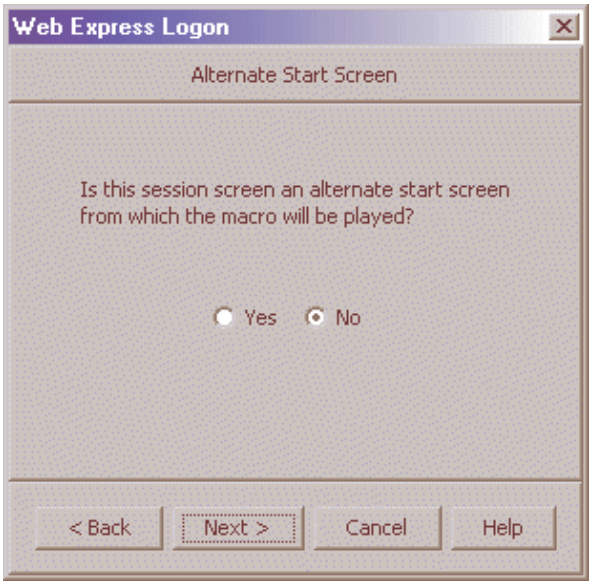
D. Enter the application ID (3270 sessions only) in the Application ID window and then click OK.

This name must match the RACF PTKTDATA (Passticket Data Profile) application name that is configured on the z/OS host. This name could be the same as the application name that the user is logging on to (for example, the name on USSMSG10). When creating PTKTDATA profiles for applications such as TSO (time sharing option), the application name portion of the profile will most likely not be the same. For example, RACF requires that the application ID portion of the profile name be TSO+SID. Refer to the RACF Security Administrator's Guide to determine the correct profile naming. You can obtain this ID from the host administrator.

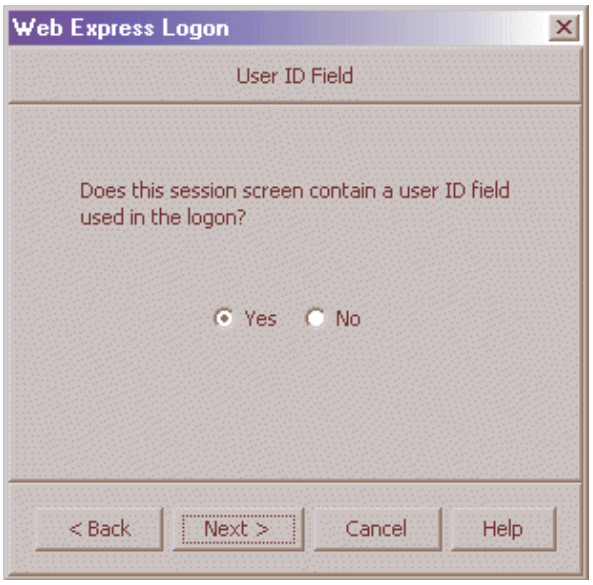
E. The Screen Criteria window shows you what is needed by the macro to complete the logon. Once you reach a screen that meets any of the criteria, click OK.

F. On the Alternate Start Screen window, specify whether this screen is an alternate start screen and click Next. The macro can start playing when a start screen is recognized or when an alternate screen is recognized. You can have only one alternate start screen per logon. If you have multiple logons, you will pass through this screen Again.

The alternate start screen is a screen from which the user might want to play the macro to log on to the application. If the application has more than one possible start screen, you should identify it during the recording process so that the macro can be played from that screen. For example, the logon process might begin from the USSMSG10 screen or the application logon screen. You may start the logon macro from either the start screen or the alternate start screen. You can designate only one screen as an alternate start screen. There is no alternate start screen after the screen that contains the user ID.

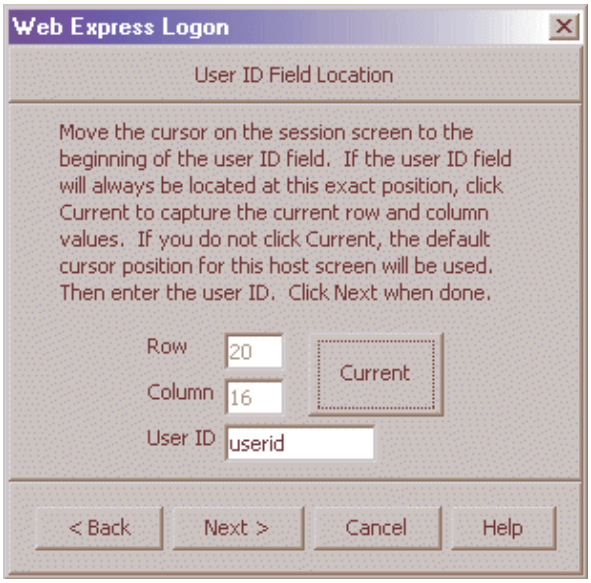


G. On the User ID Field window, select Yes to specify that the session screen contains a user ID field. Click Next.



H. On the User ID Field Location window, type the user ID in the User ID field, not on the session screen. You must enter a user ID to continue recording the macro. The macro enters the actual user ID text in the user ID field on the session screen. Row/column determines the cursor position on the screen for the user ID field. Click Current to use the cursor's current position on the session screen if you know it is correct. If the current cursor position is not correct, move the cursor to the beginning of the user ID field on the session screen to identify where the user will enter the user ID and click Current. The field values change to match the new cursor position on the screen. If the initial cursor position is correct, then there is no need to move the cursor on the session screen. When you are finished, click Next.

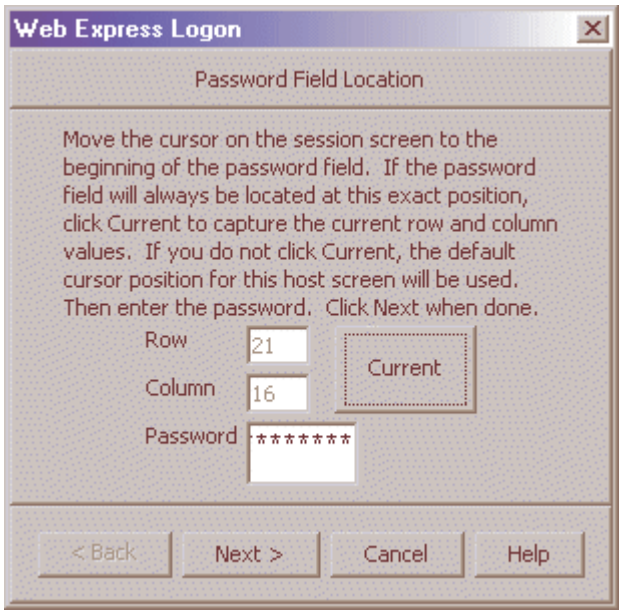
Click Current only if you will not be using this screen for multiple applications and the location of the user ID field never changes.



I. On the Password Field window, select Yes to specify that the session screen contains a password field. Click Next.

J. On the Password Field Location window, type the password in the Password field on this window, not on the screen. You must enter a password to continue recording the macro. The macro enters the actual password text in the password field on the session screen. Row/Column determines the cursor position on the screen for the password field. Click Current to use the cursor's current position on the session screen if it is correct. If not, move the cursor to the beginning of the Password field on the session screen to specify where the user will enter the password and click Current. The field values change to match the new cursor position on the screen. If the initial cursor position is correct, then there is no need to move the cursor on the session screen. When you are finished, click Next.

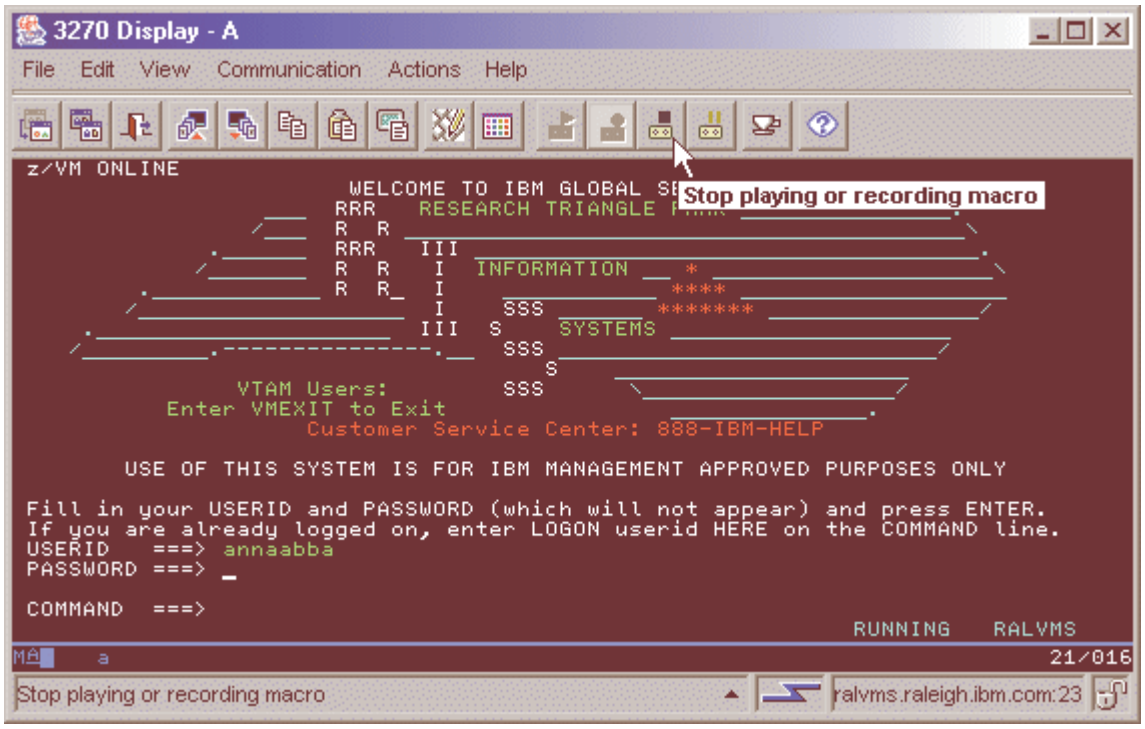
Click Current only if you will not be using this screen for multiple applications and the location of the password field never changes.



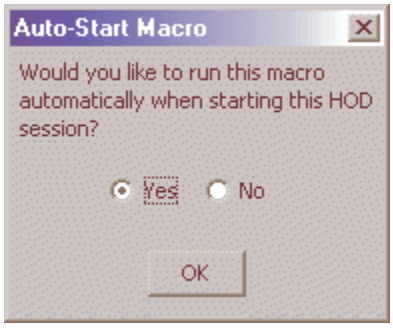
K. On the Multiple Logons window, select No and click Finish to finish recording the logon portion of the macro. Click Yes only if you you want to define another logon sequence for an additional Application.

L. Click OK to stop recording the Macro.

M. Finish recording your macro using the Macro Manager, and click the Stop recording macro button on the Toolbar.



N. If you are planning to save the macro to your current session (and not to a file), another window appears that asks you if you would like the macro to start automatically when the user opens the session. Click Yes if you would like the macro to auto-start. If you select No, the user will have to start the macro Manually.

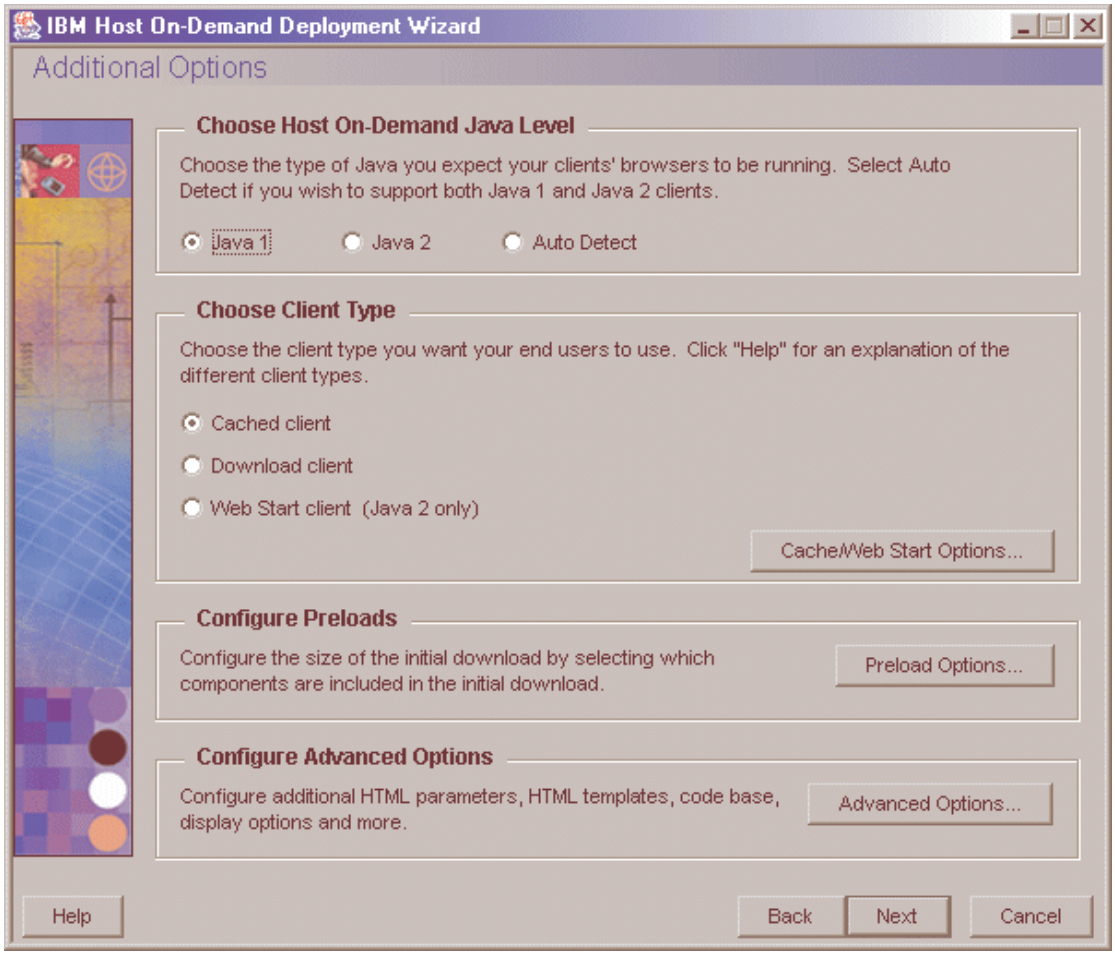


O. Close your session to return to the Host Sessions window.

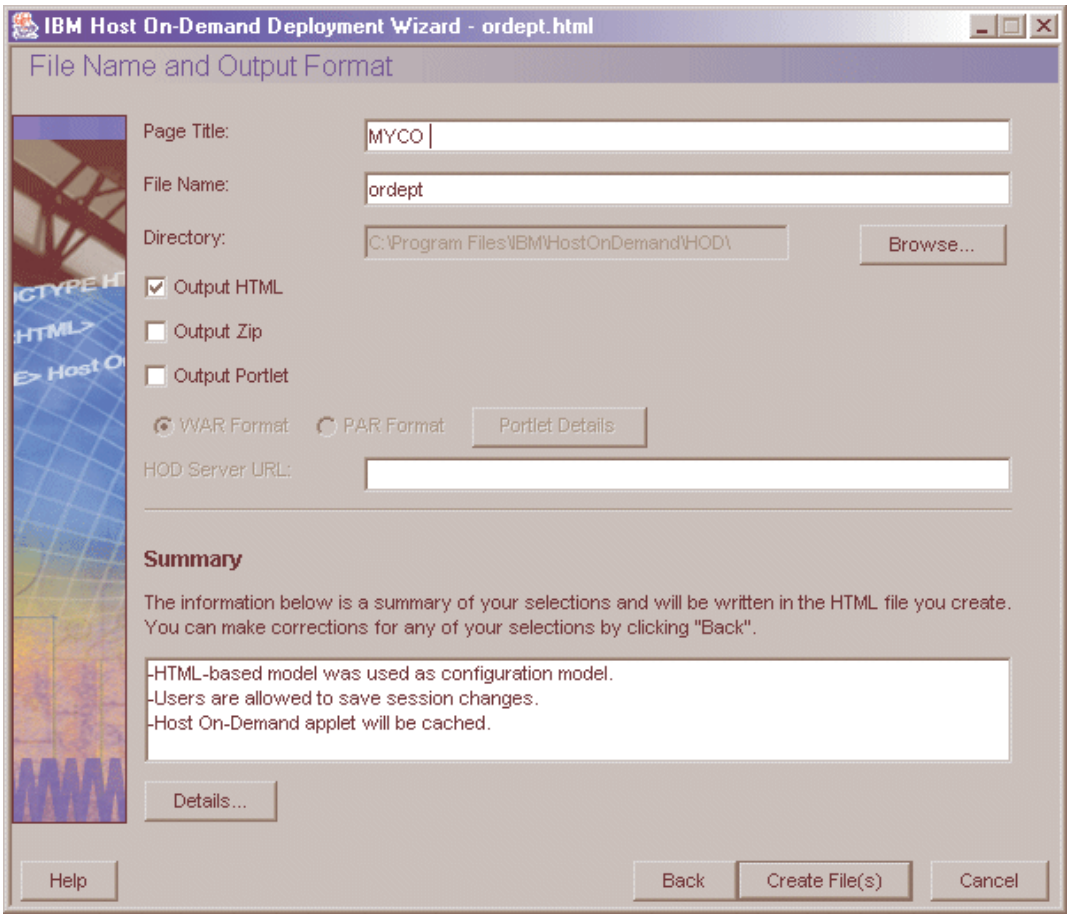
Step 9 of 9: Finish creating your HTML file.

Now that you have configured your Host On-Demand session to use Web Express Logon and have recorded your login macro, you are ready to finish creating your HTML file. At this point, you are still using the Deployment Wizard tool (continued from Step 8).

A. On the Host Sessions window, click Next to open the Additional Options window. Make any changes that you desire and click Next.



B. On the File Name and Output Format window, enter the page title, the file name, and choose the directory where you want to save your file. You should save it to the Host On-Demand server in a directory known to your Web server; usually, this directory is your Host On-Demand server's publish Directory. Click Create File(s) to finish creating your HTML file.



Congratulations! You have now completed Scenario #2: Configuring Web Express Logon in a vault-style environment. To troubleshoot any problems, refer to Troubleshooting Web Express Logon on page 131.

Scenario #3: Connection-based automation: Configuring Web Express Logon in an OS/400 and Kerberos environment

You are an administrator who manages the network for the shipping and receiving department for a large apparel manufacturer. Throughout the day, hundreds of manufacturer's representatives log on to the network and connect to two iSeries host systems (both running OS/400 V5R2) to access order entries, order status, and shipping and inventory information. With Host On-Demand, they have immediate, Web-based access to this data. You are in charge of maintaining this environment.

Now that you have upgraded to Host On-Demand Version 8 and OS/400 V5R2, you plan to accomplish two main tasks:

- enable single sign-on in the OS/400 environment so employees can use their Windows sign-on information to authenticate themselves to the OS/400 hosts without being prompted with a login screen
- implement Host On-Demand's Web Express Logon so employees can use their user IDs and tickets obtained from the Windows key distribution center (KDC) without being prompted with a login screen

Before you implement Web Express Logon, you must configure your OS/400 environment for single sign-on capability. This requires you to configure network authentication service (NAS) and Enterprise Identity Mapping (EIM), both of which are available with the OS/400 V5R2 operating system. In broad terms, NAS allows an iSeries server to participate in a Kerberos realm, and EIM provides a mechanism for associating Kerberos principal names (names of users in a Kerberos network) to a single EIM identifier that represents that user in the entire enterprise. They work together to provide a single sign-on environment. Host On-Demand uses this existing methodology for acquiring credentials to allow users to bypass the host session login screen.

In this scenario, you must configure NAS so your OS/400-based iSeries systems will accept Kerberos tickets from the Windows server KDC. The KDC maintains a database of principal names and passwords within the Kerberos realm. When users attempt to access to an application, they request a ticket called a ticket granting ticket (TGT) from the KDC. If authenticated, they are granted a TGT and can access the desired application.

OS/400 single sign-on capability can work with only one iSeries host server or on multiple iSeries systems. In this scenario, you are configuring two iSeries systems.

To configure OS/400 single sign-on and Web Express Logon, you take the following steps:

1. Complete the planning worksheets.
2. Enable OS/400 single sign-on: Part I.
3. Enable OS/400 single sign-on: Part II.
4. Begin creating your HTML file.
5. Configure your Host On-Demand session.
6. Finish creating your HTML file.

Step 1 of 6: Complete the planning worksheets.

The following prerequisite checklists illustrate the type of information you will need before you begin enabling single sign-on in your OS/400 environment.

Prerequisite checklist	Answers
Is your OS/400 V5R2 (5722-SS1) or later?	Yes
Is Cryptographic Access Provider (5722-AC3) installed on your iSeries systems?	Yes
Is iSeries Access for Windows (5722-XE1) installed on the PC that you will use to configure NAS?	Yes
Is the Security subcomponent of iSeries Navigator installed on the PC that you will use to configure NAS?	Yes
Is the Network subcomponent of iSeries Navigator installed on the PC that you will use to configure NAS?	Yes
Do you have *SECADM, *ALLOBJ, and *IOSYSCFG special authorities?	Yes
Is your system value set to *VERIFY? To change the value, use either the iSeries command line or iSeries Navigator. Using the iSeries command line, take the following steps: 1. Type the following command: `WRKSYSVAL SYSVAL(QRMTSIGN)` 2. Enter the number 5 to display your current system value. If this value is `*FRCSIGNON`, enter the number 2 and change it to `*VERIFY`. To use iSeries Navigator, take the following steps: 1. From your target iSeries server, click Configuration and Service > System Values > Sign-on > Remote. 2. Under 'Use Telnet for remote sign-on', check 'Allow sign-on to be bypassed'. 3. Select 'Use Pass-through for remote sign-on'. 4. Select 'Allow sign-on to be bypassed' and then 'Verify user ID on target system'.	Yes
Have you confirmed that your iSeries software clock is synchronized with a specified time server? The Simple Network Time Protocol (SNTP) client allows you to do this. You can specify an amount of time that the iSeries software clock must be near the time server before the SNTP client will adjust the time of day on your software clock. This function is particularly important when using Network Authentication Service (NAS). In iSeries Navigator, you can start and stop your SNTP client. You can also specify the time server to compare the iSeries software clock, and select when you would like SNTP activity to be logged. To start or stop the SNTP client in iSeries Navigator, follow these steps: 1. Expand your iSeries server > Network > Servers > TCP/IP. 2. Right-click SNTP, and select Start or Stop, as appropriate. To adjust the SNTP client parameters in iSeries Navigator, follow these steps: 1. Expand your iSeries server > Network > Servers > TCP/IP.	Yes

2. Right-click SNTP, and select Properties to display the SNTP Properties pages. 3. Adjust parameters in the General and Additional parameters tabs. 4. For additional information, click the Help button on the General and Additional parameters tabs. 5. Click OK. Note: The remote time server host must be configured before the SNTP client can start.	
Do you have one of the following installed on the secure system that will act as the KDC? If so, which one? Windows 2000 or Windows 2003 Server AIX Server zSeries	Yes Windows 2000 Server
For Windows 2000 Server and Windows XP Server, do you have Windows Support Tools, which provides the ktpass tool, installed on the system being used as the key distribution center?	Yes
Are all your PCs in your network configured in a Windows 2000 domain?	Yes
Have you applied the latest program temporary fixes (PTFs)? (The latest PTFs are located on the IBM eServer iSeries support site at http://www.ibm.com/servers/eserver/support/iseries/	Yes
Is the iSeries system time within five minutes of the KDC's system time?	Yes

You need this information to configure NAS	**Answers**
What is the name of the Kerberos default realm to which iSeries-A and iSeries-B will belong?	ORDEPT.MYCO.COM
What is the KDC for this Kerberos default realm? What is the port on which the KDC listens?	kdc1.ordept.myco.com 88
Do you want to configure a password server for this default realm? If yes, answer the following questions: What is name of the password server for this KDC? What is the port on which the password server listens?	YES Kdc1.ordept.myco.com 464
What is the host name of the iSeries servers on which you are configuring NAS?	iSeries-A and iSeries-B
What is the password for your iSeries service principal(s)?	iseriesa123
What additional realms will your iSeries systems interact with?	N/A
For each realm, what is the host name of the KDC?	N/A

You need this information to configure EIM	**Answers**
What is the host name of the iSeries server on which you are configuring EIM?	iSeries-B
What is the LDAP administrator's distinguished name and password?	distinguished name: cn=administrator password: mycopwd
What is the name if the Directory Services (LDAP) server?	iseriesb.ordept.myco.com
What is the port number of the Directory Services (LDAP) server?	389

Step 2 of 6: Enable OS/400 single sign-on: Part I.

The first part of enabling OS/400 single sign-on is configuring network authentication service (NAS) on both iSeries-A and iSeries-B. You will configure Enterprise Identity Mapping (EIM) in the next step.

You must configure NAS on all iSeries servers in your network that will participate in the Kerberos realm, including the server that will serve as the EIM domain controller. If there is only one iSeries server in your environment, you can configure both NAS and EIM on one server.

To configure NAS on iSeries-A and iSeries-B, the administrator takes the following steps:

A. Verify host name and TCP/IP domain information.
B. Configure NAS on iSeries-A.
C. Add iSeries-A principal name to the KDC.
D. Create a home directory for each user on iSeries-A.
E. Test NAS configuration on iSeries-A.
F. Repeat steps 2-5 on iSeries-B.

Any and all passwords used within this scenario are for example purposes only. They should not be used during an actual configuration.

****

A. Verify host name and TCP/IP domain information: In order to enable single sign-on capability, all components within the network must agree on the same host name and TCP/IP domain information. Otherwise, the components may not be able to "talk" to each other and users will not be properly authenticated. In this step, you will confirm that your PC and iSeries servers agree on the same information. Performing this step before you begin configuring NAS may save you valuable time in later steps.

Take the following three steps, i-iii:

i. From your Windows PC, determine the fully qualified TCP/IP host name for the OS/400 server.

Depending on how you manage your network, you may wish to do this on other PCs that take part in the single sign-on environment.

(1) Open the hosts file. This file contains the mappings of IP addresses to host names. Take note of the system name of the first entry, for example, iseriesa.ordept.myco.com. Also note the upper and lower case characters. The path to this file depends on your Windows operating system:

- Windows 2000 operating system: C:\WINNT\system32\drivers\etc\hosts
- Windows XP operating system: C:\WINDOWS\system32\drivers\etc\hosts

If the hosts file does not exist on your PC or there is no OS/400 server entry in your hosts file, then your PC may be using a DNS server to resolve host names. If this is the case, proceed with the rest of Step i. Otherwise, skip to Step ii.

(2) If you either did not find the hosts file on your PC or your hosts file does not contain an entry for the OS/400 server, use NSLOOKUP (Name Server Lookup) to query the DNS server for the host name and IP address.

a. At a command prompt, type NSLOOKUP and press Enter. At the NSLOOKUP prompt, type the IP address of the host and press Enter. The DNS server returns the host name of the OS/400 server. Take note of the name, including the upper and lower case characters. In this scenario, the DNS server returned iseriesa.ordept.myco.com.

b. At the NSLOOKUP prompt, type the name of the host name that was returned by the DNS server (iseriesa.ordept.myco.com) and press Enter. Verify that the DNS server returns the IP address that you expect.

Important: If NSLOOKUP does not return the expected results, you have an incomplete DNS record. For example, if NSLOOKUP returns an IP address that is different than the address you entered in this step, you need to contact the DNS administrator to resolve this problem before you can continue with the next steps.

ii. From the OS/400 server, determine the fully qualified TCP/IP host name.

(1) Determine TCP/IP domain information:

a. At the command prompt, type CFGTCP and select Option 12 (Change TCP/IP domain information).
b. Take note of the values for the Host name parameter and the Domain name parameter, including the upper and lower case characters. In this scenario, the host name is iseriesa and the domain name is ordept.myco.com.
c. Take note of the value for the Host name search priority parameter. This value depends on how the administrator configured TCP/IP to perform host resolution on the server.

- ***LOCAL**: The operating system searches the local host table (equivalent to the hosts file on the PC) first. If no matching entry exists in the host table and you have configured a DNS server, the operating system searches your DNS server.
- ***REMOTE**: The operating system searches the DNS server first. If no matching entry exists in the DNS server, the operating system searches the local host table.

(2) Determine TCP/IP host table information:

a. At the command prompt, type CFGTCP and select Option 10 (Work with TCP/IP host table entries).
b. Take note of the value in the Host Name column that corresponds to the OS/400 server, including the upper and lower case characters. In this scenario, the value is iseriesa.ordept.myco.com.

If you do not find an entry for the OS/400 server in the host table, proceed to the next step.

(3) Determine DNS server information:

a. At a command prompt, type NSLOOKUP and press Enter. At the NSLOOKUP prompt, type the IP address of the host and press Enter. The DNS server returns the host name of the OS/400 server. Take note of the name, including the upper and lower case characters. In this scenario, the DNS server returned iseriesa.ordept.myco.com.

b. At the NSLOOKUP prompt, type the name of the host name that was returned by the DNS server (iseriesa.ordept.myco.com). Verify that the DNS server returns same IP address that you expect.

Important: If NSLOOKUP does not return the expected results, you have an incomplete DNS record. For example, if NSLOOKUP returns an Internet address that is different than the address you entered in this step, you need to contact the DNS administrator to resolve this problem before you can continue with the next steps.

(4) Determine which host name value for the OS/400 server to keep, based on the TCP/IP configuration.

- If the value for the Host name search priority parameter is *LOCAL, keep the entry noted from the local host table (Step ii - (2)b)
- If the value for the Host name search priority parameter is *REMOTE, keep the entry noted from the DNS server (Step ii - (3)a)
- If only one of these sources contains an entry for the OS/400 server, keep that entry.

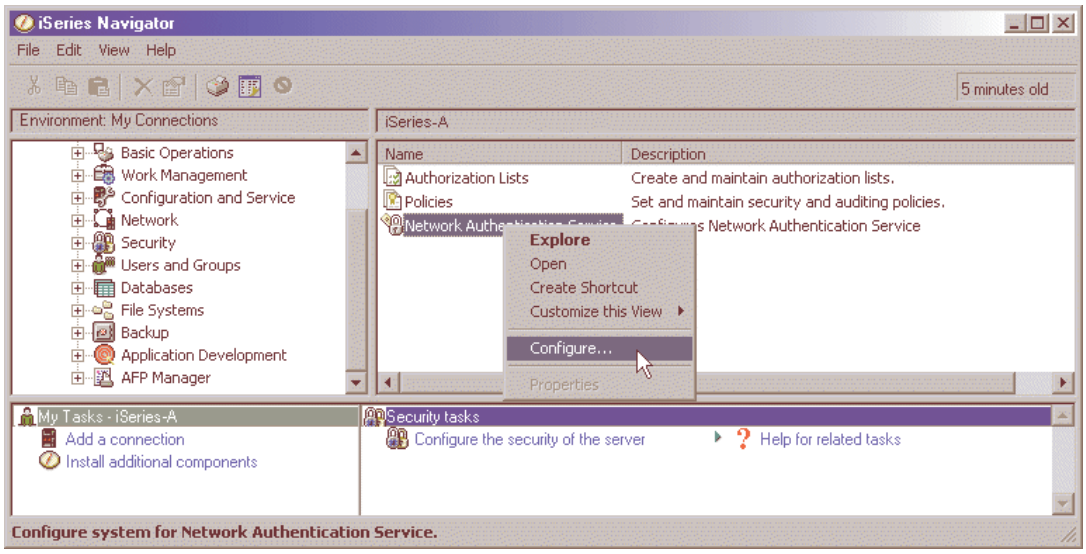
iii. Compare the results from the following steps:

- Step i: Name that the PC uses for the OS/400 server (If you found an entry for the OS/400 in the PC's hosts file, use that entry. Otherwise, use the entry from the DNS server.
- Step ii - (2)b: Name that the OS/400 server calls itself within the TCP/IP configuration.
- Step ii - (4): Name that the OS/400 server calls itself based on the host name resolution.

All three of these entries must match exactly, including upper and lower case characters. If the results do not match exactly, you will receive an error message indicating that a keytab entry cannot be found.

B. Configure NAS on iSeries-A: Use the information from your worksheets to configure NAS on iSeries-A by completing the following tasks:

i. Open iSeries Navigator and expand iSeries-A > Security. Right-click Network Authentication Service and select Configure to start the configuration wizard.



Once you have configured NAS, this option changes to Reconfigure.

ii. Review the Welcome page for information about what objects the wizard creates. Click Next.
iii. On the Specify Realm Information page, enter ORDEPT.MYCO.COM (in all upper case) in the Default realm field. Click Next.
iv. On the Specify KDC Information page, enter kdc1.ordept.myco.com in the KDC field and accept the default 88 in the Port field. Click Next.
v. On the Specify Password Information page, select Yes. Enter kdc1.ordept.myco.com in the Password server field and 464 in the Port field. Click Next.
vi. On the Create Keytab Entry page, select iSeries Kerberos Authentication. Click Next.
vii. On the Create iSeries Keytab Entry page, write down the keytab and principal for iSeries-A. You will need the principal name when you add this to the KDC. Enter and confirm a password. For example, the administrator for MyCo entered iseriesa123. Click Next.
viii. On the Summary page, review the NAS configuration details. Click Finish.

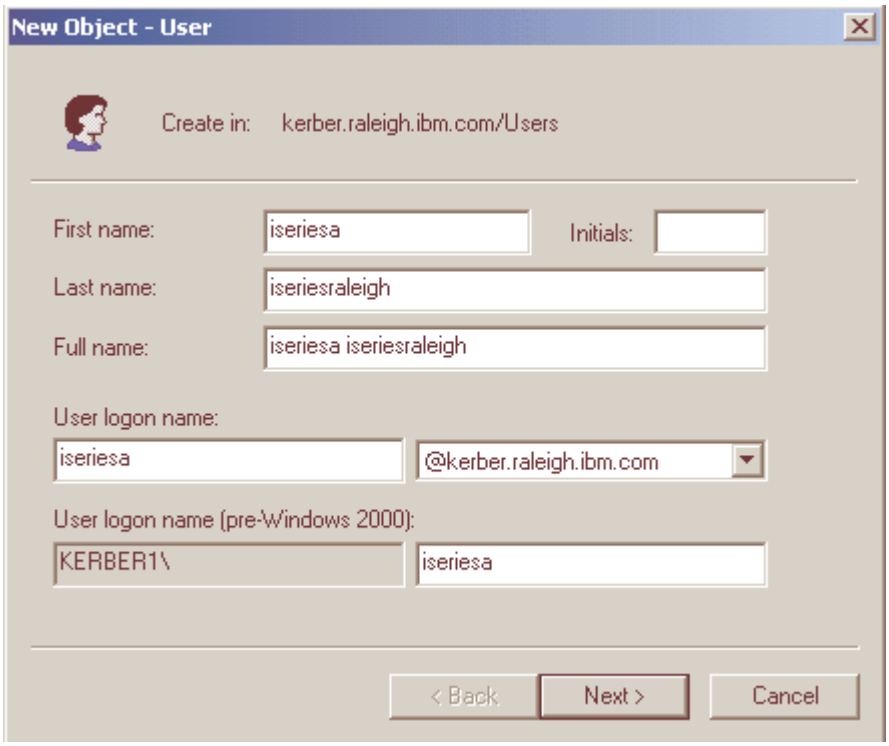
Now you are finished configuring NAS on iSeries-A. The next step is to add the principal name to the KDC.

C. Add iSeries-A service principal name to the KDC: To add the iSeries system to the Windows 2000 KDC, you must first use Microsoft Active Directory (R) to create accounts for your network users and then map these users to the Kerberos service principal. By convention, the iSeries host name (iseriesa.ordept.myco.com) can be used as the username. Take these steps on your Windows 2000 server to add the following service principal name to the KDC:

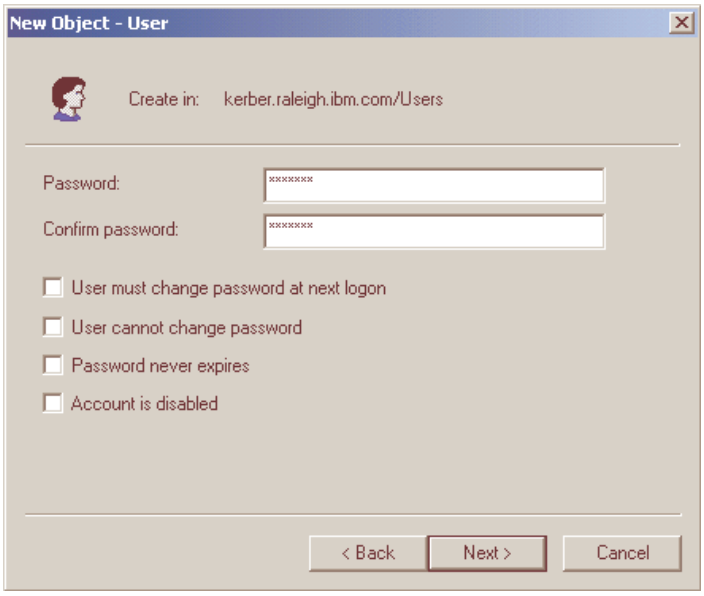
krbsvr400/iseriesa.ordept.myco.com@ORDEPT.MYCO.COM

where krbsvr400 is the name of the service, iseriesa.ordept.myco.com is the host name of the iSeries machine, and ORDEPT.MYCO.COM is the name of the realm. Service principal names are made up of these three parts.

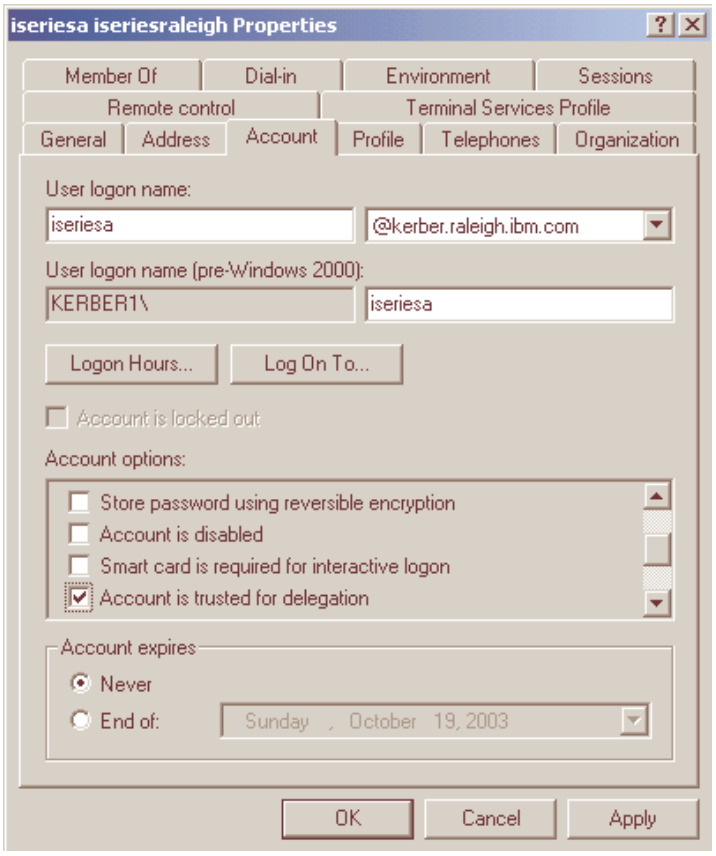
i. Use the Active Directory Management tool to create a user account for the iSeries system (select the Users folder, right-click, select New, then select User.) Specify the first and last name of the Active Directory user account. In this scenario, the administrator types iseriesa for first name, iseriesaraleigh as the last name, and iseriesa as the user logon name. Click Next.



ii. Type the password in the Password and Confirm password fields. Click Next and then Finish.



iii. Access the properties for the Active Directory user iseriesa by double-clicking the Active Directory user name in the user list window. From the Account tab, select the Account is trusted for delegation. This allows the iSeries-A service principal to access other services on behalf of a signed-in user.



iv. From the KDC server, map the iSeries user account to the principal by using the ktpass command. The ktpass tool is provided in the Service Tools folder on the Windows 2000 Server installation CD. To map the user account, enter the following:

```
ktpass -princ krbsvr400/iseriesa.ordept.myco.com@ORDEPT.MYCO.COM
-mapuser iseriesa -pass iseriesa123 -mapop set
```

where iseriesa123 is the password that you specified when you configured NAS in the Step #2, part G. Adding -mapop set to the command is optional but is recommended because it removes any existing mappings.

D. Create a home directory for each user on iSeries-A: Each user that connects to the iSeries server and iSeries applications needs a subdirectory in the /home directory. This directory contains the name of the user's Kerberos credentials cache. To create a home directory for a user, complete the following steps:

On an iSeries command line, enter the following:

```
CRTDIR 'home/username '
```

where *username* is the user's iSeries username. For example, the administrator for MyCo entered CRTDIR 'home/Johns ' for the user John Smith.

Repeat these steps for all your network users.

E. Test NAS configuration on iSeries-A: At this point, you can verify that you have configured NAS correctly by requesting a ticket granting ticket for iSeries-A principal name.

To perform this step, you must have created a home directory (see Step D); otherwise you cannot run this command.

i. On a command line, enter QSH to start the Qshell Interpreter.
ii. Enter keytab list to display a list of principals registered in the keytab file. In this scenario, krbsvr400/iseriesa.ordept.myco.com@ORDEPT.MYCO.COM should display as the principal name for iSeries-A.

If you chose to configure principals for LDAP and iSeries NetServer, there will be other entries in the keytab file. In this scenario, the administrator chose not to configure principals for these services.

iii. Enter kinit -k krbsvr400/iseriesa.ordept.myco.com@ORDEPT.MYCO.COM . If this is successful, then the QSH command will display without errors.
iv. Enter klist to verify that the default principal is krbsvr400/iseriesa.ordept.myco.com@ORDEPT.MYCO.COM.

F. Repeat steps 2- 5 for iSeries-B.

To troubleshoot NAS, go to http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html. Once you click your region, click the icon for you language and V5R2 to open the InfoCenter. On the left navigation bar, click Security > Network authentication service > Troubleshoot network authentication service.

Step 3 of 6: Enable OS/400 single sign-on: Part II.

Now that you have configured NAS, you are ready to configure EIM. Together, NAS and EIM provide a single sign-on environment.

EIM is a mechanism for mapping (associating) a person or entity to the appropriate user identities in various user registries throughout the enterprise. EIM provides APIs for creating and managing these identity mapping relationships, as well as APIs that applications use to query this information.

To configure EIM, take the following steps:

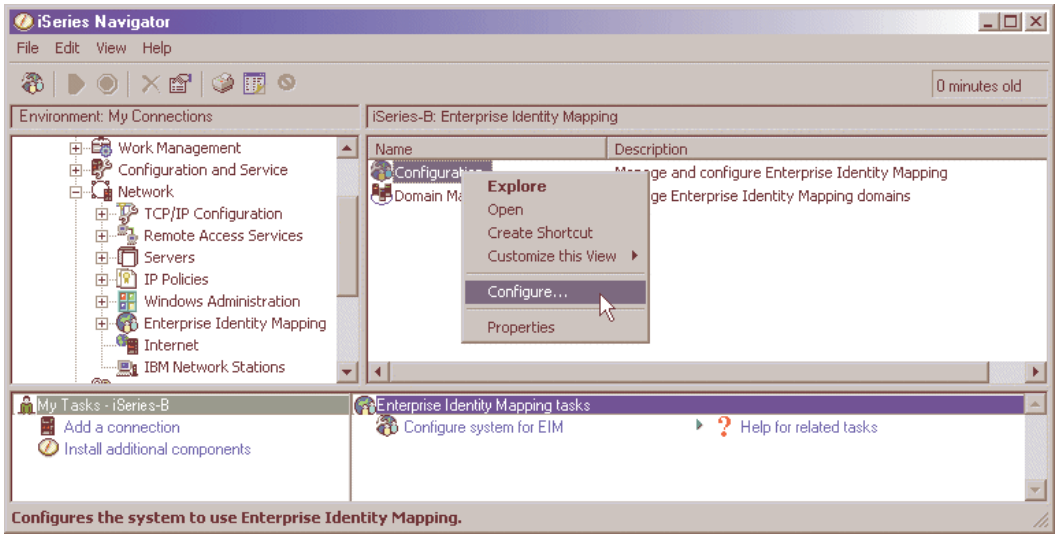
A. Configure the EIM domain and configure the directory server on iSeries-B to be the EIM domain controller.
B. Configure iSeries-A to participate in the EIM domain.
C. Create EIM identifiers for users in the enterprise.
D. Add EIM associations for the OS/400 user profiles and principal names to the EIM identifier.
E. Configure iSeries Access for Windows connections to use Kerberos principals as authentication method.
F. Verify NAS and EIM setup.

****

A. Configure the EIM domain and configure Directory Server on iSeries-B to be the EIM domain controller. You now need to configure an EIM domain in your network. You also need to configure iSeries-B to be the EIM domain controller for the new EIM domain. When you have finished this step, you will have completed the following tasks:

- Created a new EIM domain.
- Configured the Directory Server on iSeries-B to be the EIM domain controller.
- Created EIM registries for iSeries-B and Kerberos user registry in the domain.
- Configured iSeries-B to participate in the EIM domain.

i. In iSeries Navigator, expand iSeries-B > Network > Enterprise Identity Mapping.
ii. Right-click Configuration and select Configure to start the configuration wizard.



iii. On the Welcome page, select Create and join a new domain. Click Next.
iv. On the Configure Directory Server page, in the Port field, accept the default 389. In the Distinguished name field, enter cn=administrator. Enter and confirm a password. This password will be used when accessing EIM domain management tasks. For example, the administrator for MyCo entered mycopwd in the Password and Confirm password fields. Click Next.
v. On the Specify Domain page, enter the name of the domain. For example, the administrator for MyCo entered mycoeimDomain in the Domain field.

The domain name cannot contain any of the following characters: =+<>,#;\and*.

The Description field is optional. If you want, enter a brief description of the domain controller. Click Next.
vi. On the Specify Parent DN for Domain page, select No to allow EIM data to reside in its own suffix in the namespace. Click Next.
vii. On the Registry Information page, select Local OS/400 and Kerberos. Select Kerberos user identities are case sensitive. Click Next. Write down the registry names. You will need these registry names when creating associations to EIM identifiers.

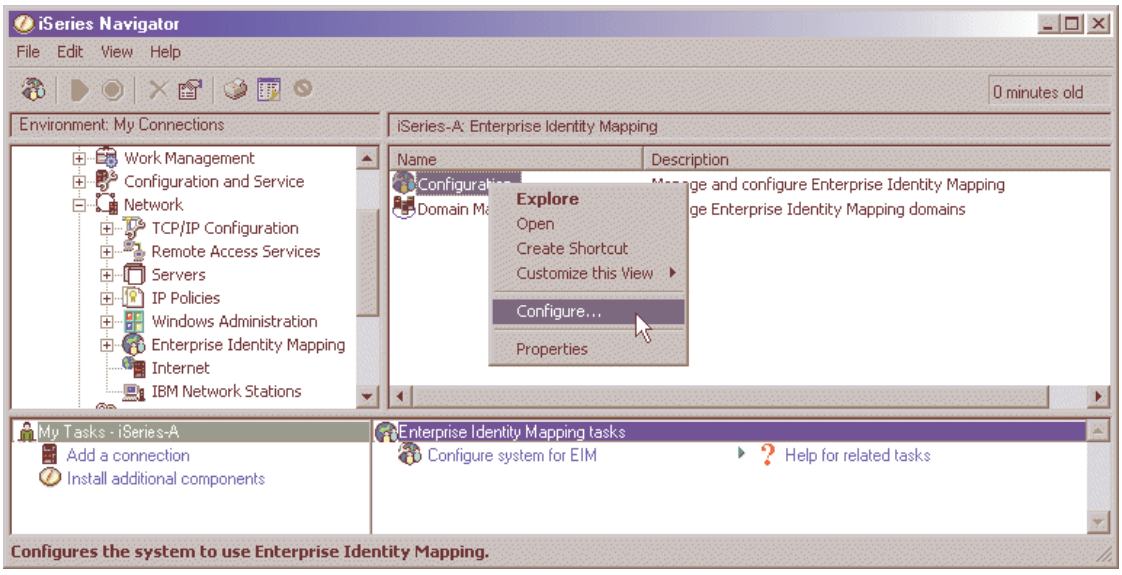
Registry names must be unique to the domain.

viii. On the Specify EIM System User page, select the system EIM user. Accept the defaults that appear on this page. For example, MyCo had the following information on this page: User type: Distinguished name and password Distinguished name: cn=administrator Password: mycopwd Click Next.
ix. On the Summary page, confirm the EIM configuration information. Click Finish.

You have now configured the directory server on iSeries-B as the EIM domain controller for the newly configured EIM domain in the network. Now you must specify iSeries-A as a participant in this EIM domain.

B. Configure iSeries-A to participate in the EIM domain: To configure iSeries-A to participate in the EIM domain, take the following steps:

i. In iSeries Navigator, expand iSeries-A > Network > Enterprise Identity Mapping.
ii. Right-click Configuration and select Configure to start the configuration wizard.



iii. On the Welcome page, select Join an existing domain. Click Next.
iv. On the Specify Domain Controller page, enter the name of the domain controller. For example, the administrator for MyCo entered iseriesb.ordept.myco.com in the Domain controller name field. Click Next.
v. On the Specify User for Connection page, select Distinguished name and password for the user type. For example, the administrator for MyCo entered cn=administrator in the Distinguished name field and mycopwd in the password and confirm password fields. Click Next.
vi. On the Specify Domain page, select the name of the domain in which you want to participate. Click Next. For example, the administrator for MyCo selected mycoeimDomain.
vii. On the Registry Information page, select Local OS/400. Click Next. Write down the registry names. You will need these registry names when creating associations to EIM identifiers.

Registry names must be unique to the domain.

viii. On the Specify EIM System User page, select the system EIM user. Accept the defaults that appear on this page. For example, MyCo had the following information on this page:

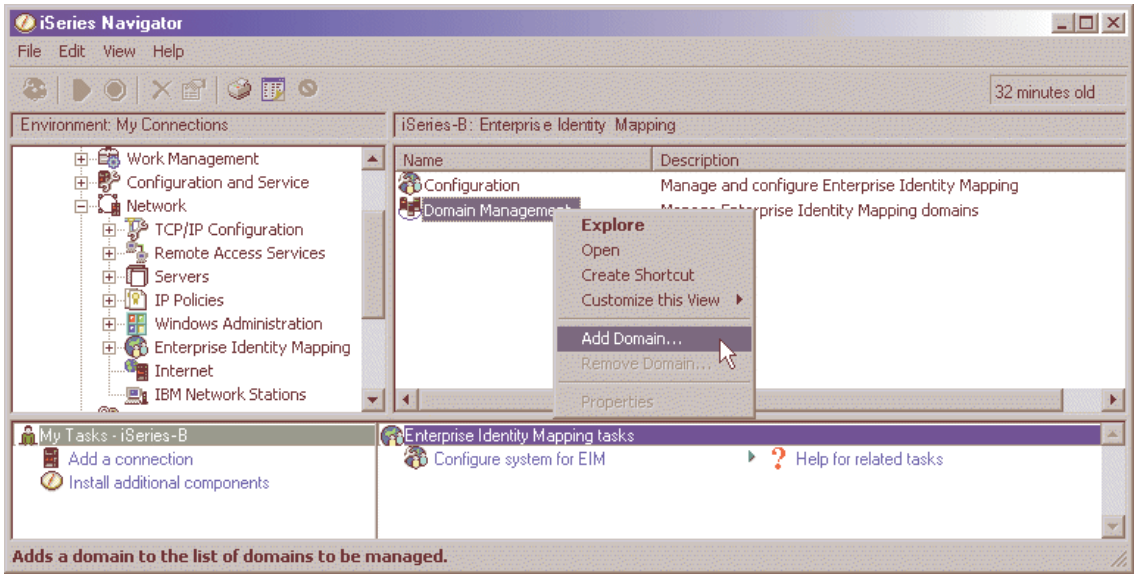
- User type: Distinguished name and password
- Distinguished name: cn=administrator
- Password: mycopwd

Click Next.
ix. On the Summary page, confirm the EIM configuration. Click Finish.

You have now configured iSeries-A to participate in the domain. You now need to create EIM identifiers for each user in the enterprise.

C. Create EIM identifiers for users in the enterprise: An EIM identifier represents a user or entity on the network. In the case of MyCo, the administrator created two EIM identifiers, John Smith and Sharon Jones.

i. On iSeries-B, expand Network > Enterprise Identity Mapping.
ii. Right-click Domain Management and select Add Domain...



iii. On the Add Domain dialog, these defaults should display for MyCo's EIM domain:

- Domain: mycoeimDomain
- Parent DN: none
- Domain controller: iseriesb.ordept.myco.com
- Port: 389

These defaults were created during EIM domain controller configuration.

Click OK.

iv. The iSeries Navigator hierarchy refreshes with mycoeimDomain under Domain Management. Click mycoeimDomain. You will be prompted with the Connect to EIM Domain Controller dialog. You must connect to the EIM domain controller before you can manage the domain.
v. On the Connect to EIM Domain Controller page, enter the Domain Controller's administrator distinguished name and password. These are the same distinguished name and password that are created during the configuration of the EIM domain controller. For MyCo, the administrator entered the following:

- Distinguished name: cn=administrator
- Password: mycopwd

vi. Click OK. Two new folders will display: User Registries and Identifiers. Right-click Identifiers and select New Identifier.
vii. On the New EIM Identifier page, enter an identifier in the Identifier field. Repeat this step until all users have an identifier. MyCo added the following identifiers:

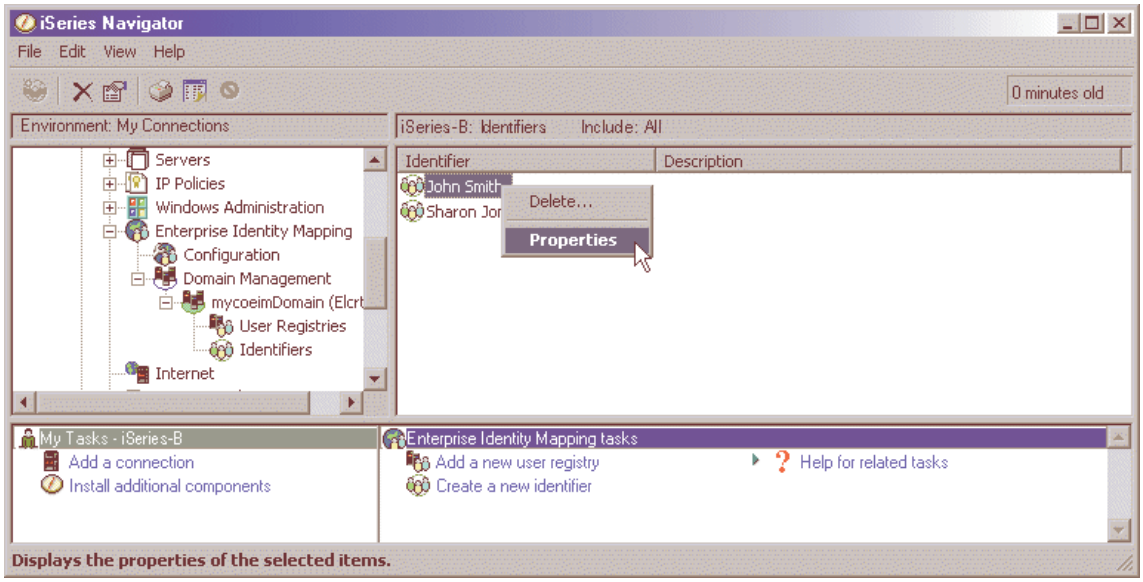
- John Smith
- Sharon Jones

viii. Click OK.

Now that unique EIM identifiers have been created for John Smith and Sharon Jones, we can now associate their OS/400 user names on iSeries-A and iSeries-B and their Kerberos principals to these EIM identifiers.

D. Add EIM associations for the OS/400 user profiles and principal names to the EIM identifier: To complete this task, MyCo's administrator completed the following steps:

i. On iSeries-B, expand Network > Enterprise Identity Mapping > Domain Management > mycoeimDomain > Identifiers.
ii. Right-click John Smith, and select Properties.



There will be three associations for this identifier: Kerberos principal, the user profile on iSeries-A, and the user profile for iSeries-B.

iii. To associate the Kerberos principal with the identifier John Smith, take the following steps:

(1) On the Associations tab, click Add.
(2) On the Add Association page, click Browse in the Registry field, and select ORDEPT.MYCO.COM. This is the Kerberos user registry that was added during EIM configuration.
(3) In the User field, enter jsmith.
(4) In the Association type field, select Source.
(5) Click OK.

iv. To associate the user name on iSeries-A with the identifier John Smith, take the following steps:

(1) On the Associations tab, click Add.
(2) On the Add Association page, click Browse in the Registry field, and select iseriesa.ordept.myco.com. This is the OS/400 user registry for iSeries-A.
(3) In the User field, enter JOHNS.
(4) In the Association type field, select Target.
(5) Click OK.

v. To associate the user name on iSeries-B with the identifier John Smith, take the following steps:

   (1) On the Associations tab, click Add.
   (2) On the Add Association page, click Browse in the Registry field and select iseriesb.ordept.myco.com. This is the OS/400 user registry on iSeries-B.
   (3) In the User field, enter Smithjo.
   (4) In the Association type field, select Target.
   (5) Click OK.

vi. Repeat these steps for user Sharon Jones.

E. Configure iSeries Access for Windows connections to use Kerberos principals as authentication method: You now need to configure both the Jsmith and Sjones PCs to use Kerberos when authenticating to the iSeries-A and iSeries-B servers. From Jsmith's PC, configure iSeries-A and its applications to use Kerberos authentication by completing the following steps:

i. In iSeries Navigator, right-click iSeries-A and select Properties.
ii. On the Connection tab, select Use Kerberos principal name, no prompting. This will allow iSeries Access for Windows connections to use the Kerberos principal name and password for authentication.
iii. Repeat these steps for iSeries-B.
iv. Repeat these steps on Sjones's PC.

F. Verify NAS and EIM setup: At this point, all configuration steps are completed. To verify that NAS and EIM have been set up correctly, the administrator had Sharon Jones and John Smith log on to the Windows 2000 domain and request access to an application on the iSeries host. If no iSeries sign-on prompt displays, EIM successfully mapped the Kerberos principal to an identifier on the domain.

You may also use a Windows 2000 Resource Kit Tool called Kerbtray to confirm that your Windows clients are able to retrieve Kerberos tickets from the KDC. Go to http://www.microsoft.com and search for kerbtray.exe. Follow the accompanying documentation to use the tool.

To troubleshoot EIM, go to http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html. Once you click your region, click the icon for you language and V5R2 to open the InfoCenter. On the left navigation bar, click Security > Enterprise Identity Mapping (EIM) > Troubleshoot EIM.

Step 4 of 6: Begin creating your HTML file.

The Host On-Demand Deployment Wizard allows you to create an HTML file that is used to launch Host On-Demand sessions. Within the Deployment Wizard, you can add, delete, configure, and start sessions. It guides you configuration choices and provides comprehensive help for the features. When you have finished selecting features, it creates the HTML and supporting files for you.

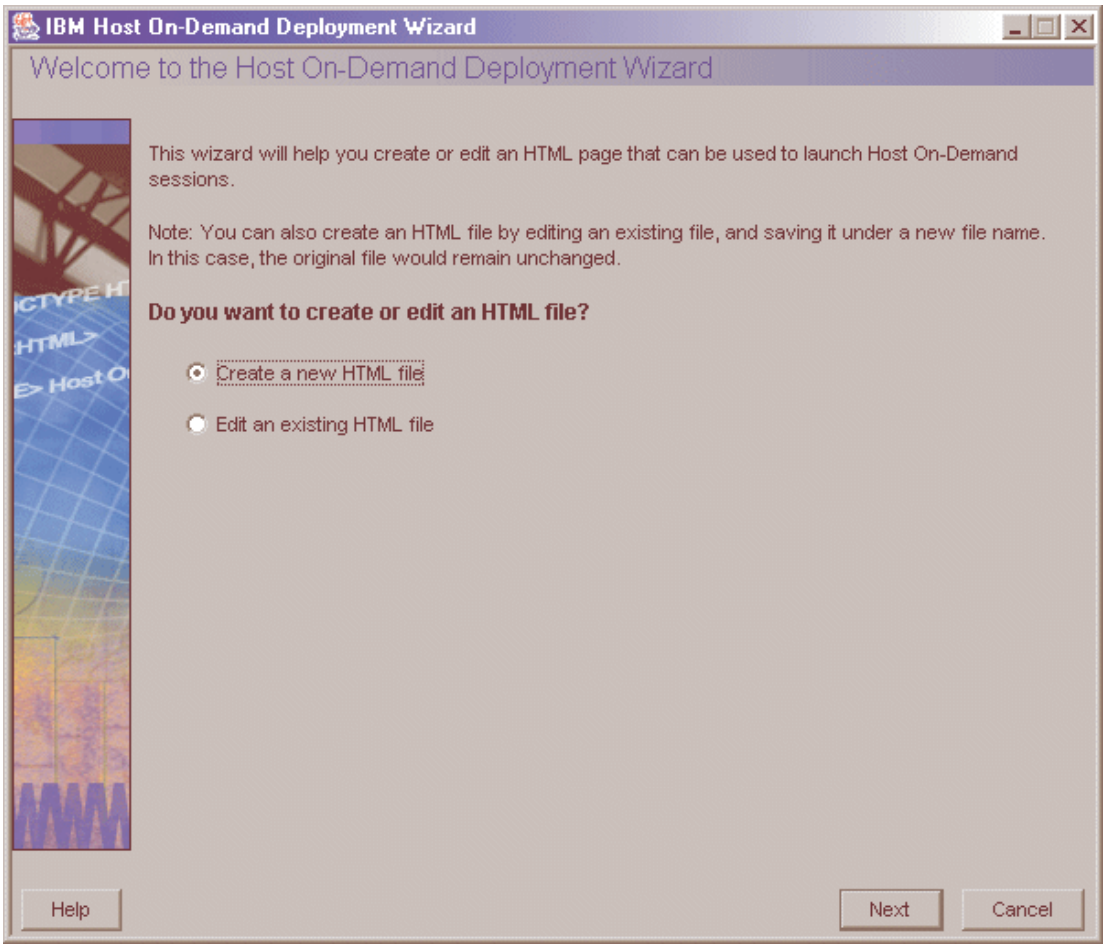
In this scenario, the administrator performs Steps 4 - 6 all within the Deployment Wizard in one sitting. However, you may decide to create your HTML file first and then configure your session later. Refer to the supplemental notes at the beginning of Step 5 for more information about how to do this.

To begin creating your HTML file, take the following steps:

A. Click Start > Programs > IBM WebSphere Host On-Demand > Administration > Deployment Wizard.

- If you automatically installed the Deployment Wizard as part of the Windows Host On-Demand server, click Start > Programs > IBM WebSphere Host On-Demand > Administration > Deployment Wizard.
- If you installed the Deployment Wizard from the Host On-Demand CD separately, click Start > Programs > IBM WebSphere Host On-Demand Deployment Wizard > Deployment Wizard.

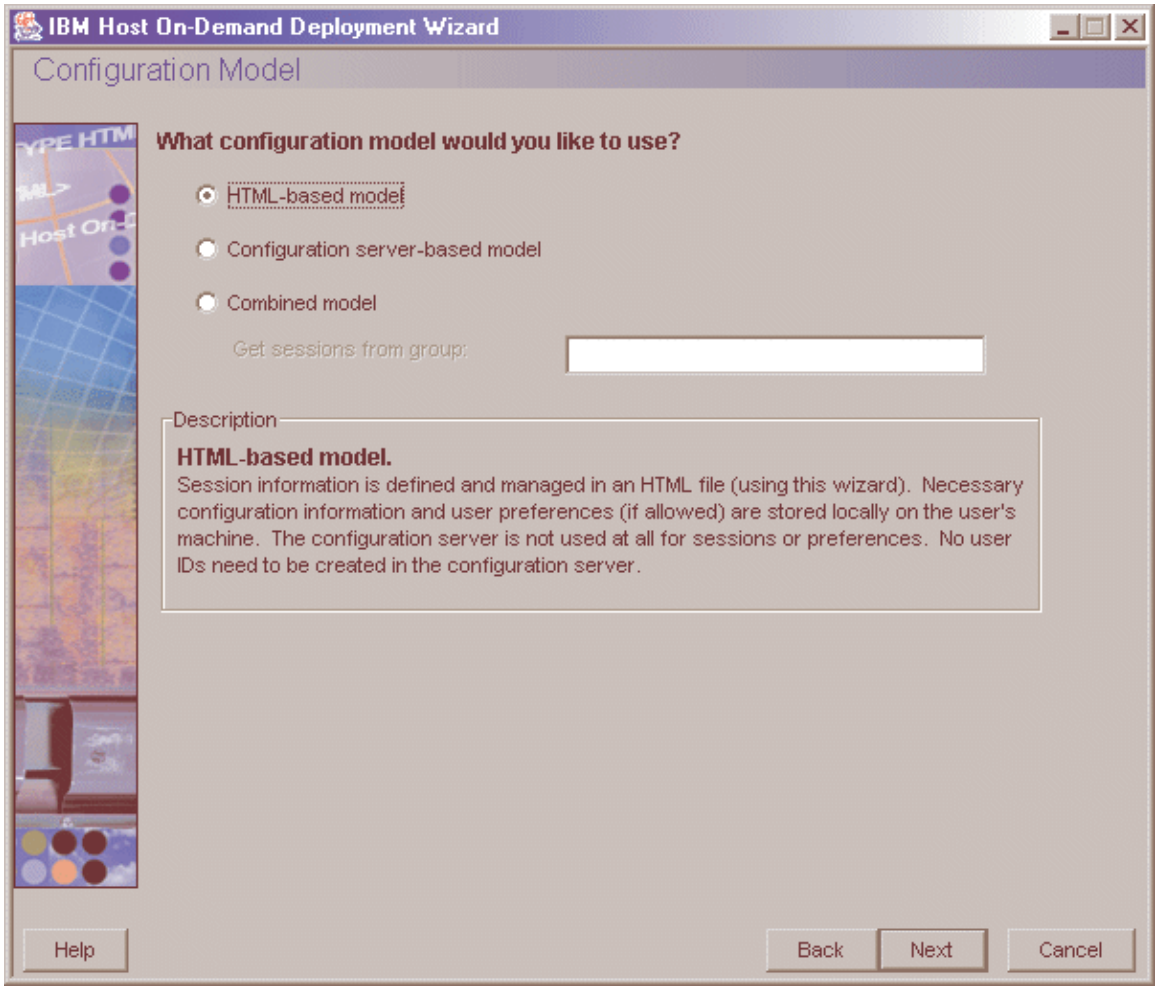
B. Select either to create a new HTML file or edit an existing file. Click Next.



C. Select from the following three configuration models and click Next:

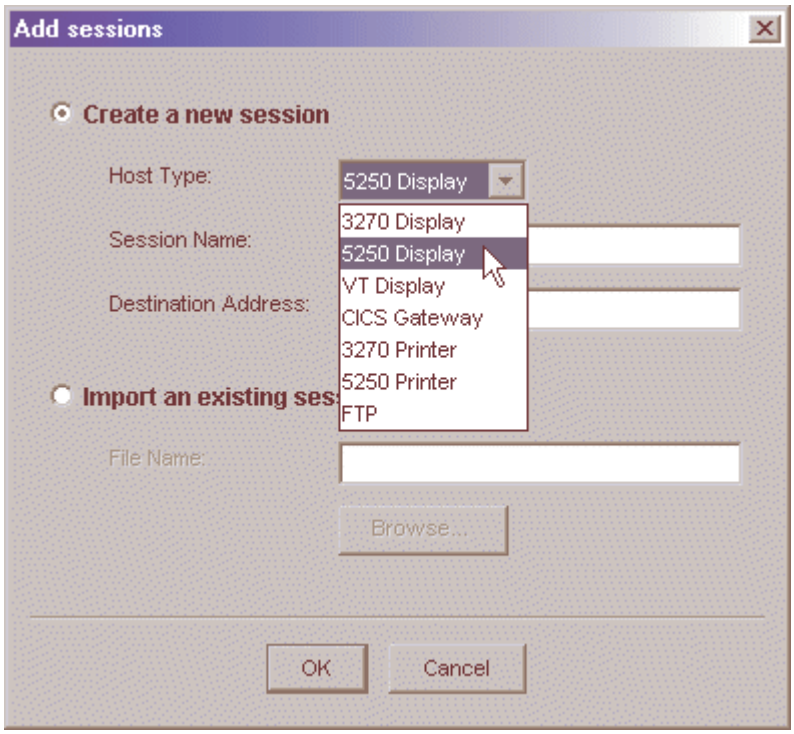
- HTML-based model
- Configuration server-based model *(*if you choose this configuration model, refer to 121.)
- Combined model

The administrator in this scenario selects the HTML based-model.



On the Host Sessions window, click New/Import to open the Add sessions window. This window allows you to either create a new session (default) or import an existing session.

To create a new session, select a host type, enter a session name, and a destination address. In this scenario, the administrator selects 5250 Display. Click OK to return to the Host Sessions window.

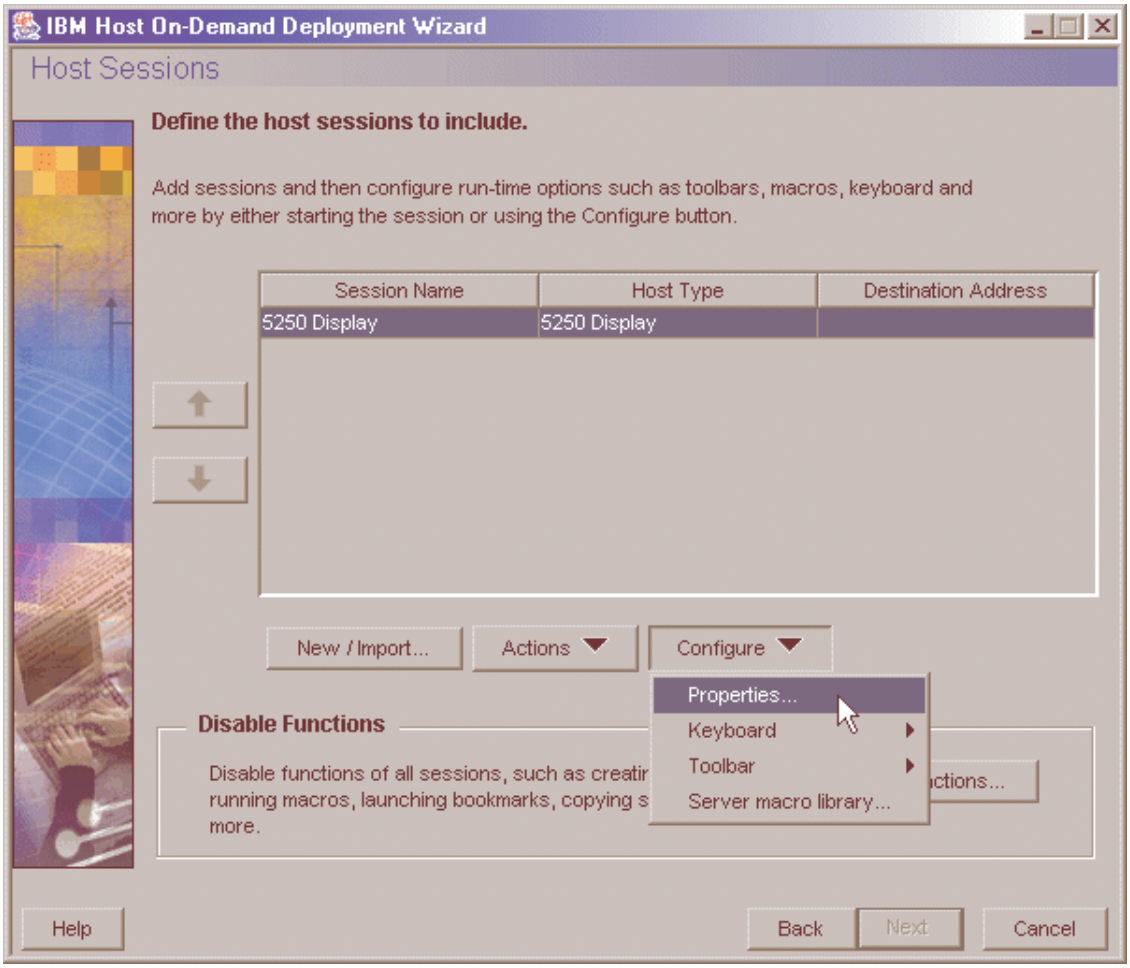


Step 5 of 6: Configure your Host On-Demand session.

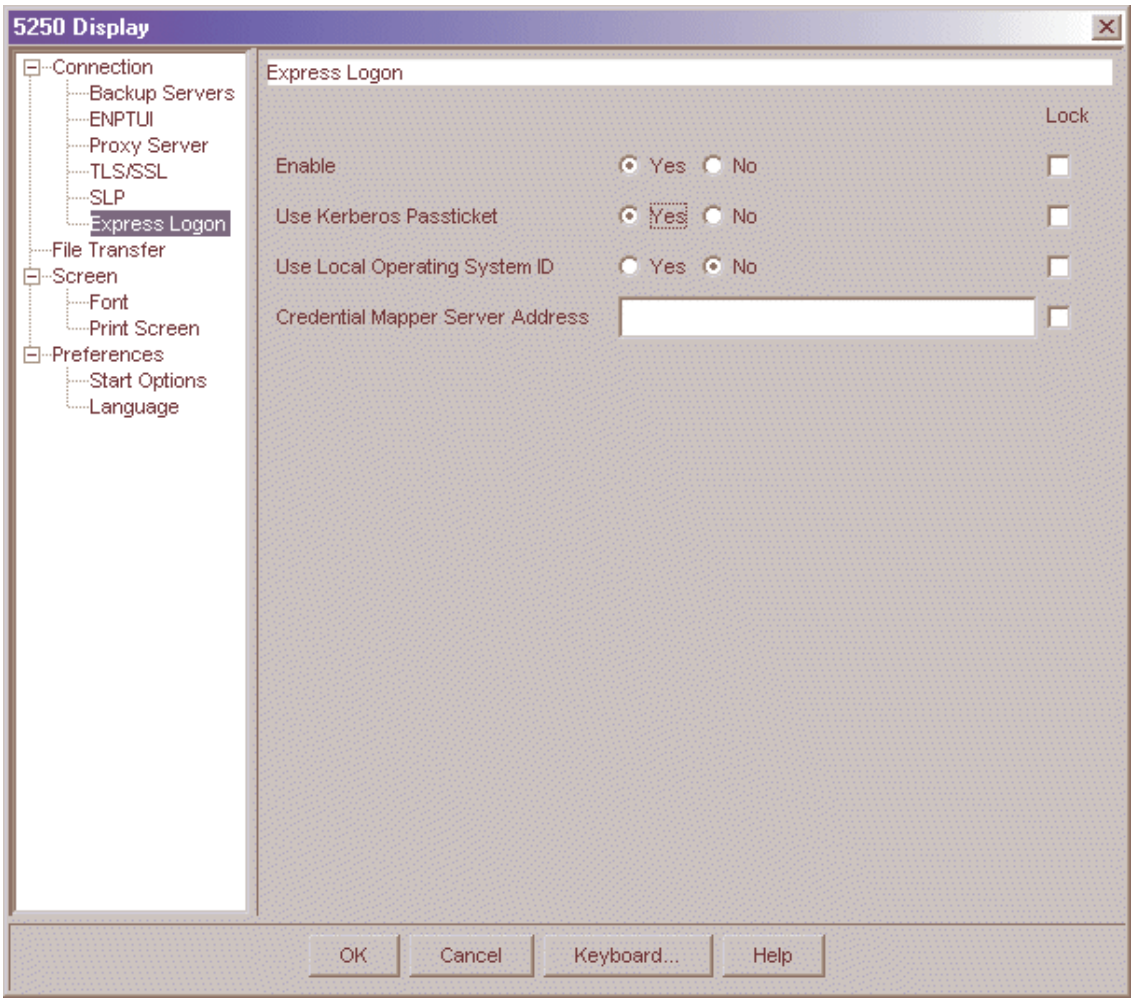
In this step, you will configure your Host On-Demand session to use Web Express Logon. At this point, you are still using the Deployment Wizard tool (continued from Step 4).

If you have already created your HTML file and now wish to configure it to use Web Express Logon, open the Host On-Demand desktop, right-click the session icon, and select Properties. Skip to Step B.

A. On the Host Sessions window, click Configure > Properties to configure your session to use Web Express Logon.



B. Under the Connection option on the left side of the 5250 Display window, click Express Logon. Select Yes to enable Express Logon and Yes to Use Kerberos Passticket.



Once you select to use a Kerberos Passticket, Host On-Demand will be able to retrieve a passticket from a Windows server. This passticket is used to connect to the host system that you identify in the session properties.

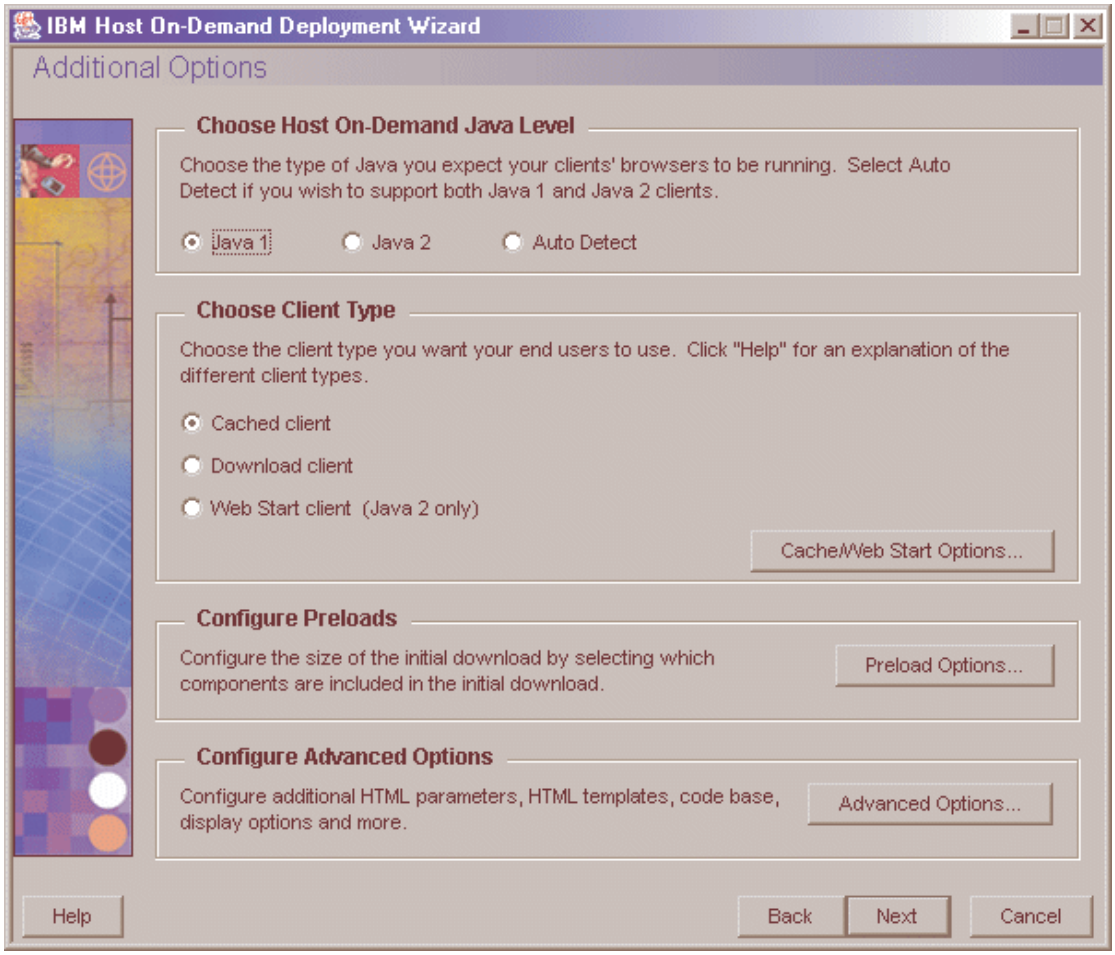
Accept the default No for Use Local Operating System ID, and leave the Credential Mapper Server Address field blank. You would only need to set these options if you had additional credential challenges that you wanted to automate through a vault-style setup. They do not apply to Kerberos authentication.

Click OK to return to the Host Sessions window.

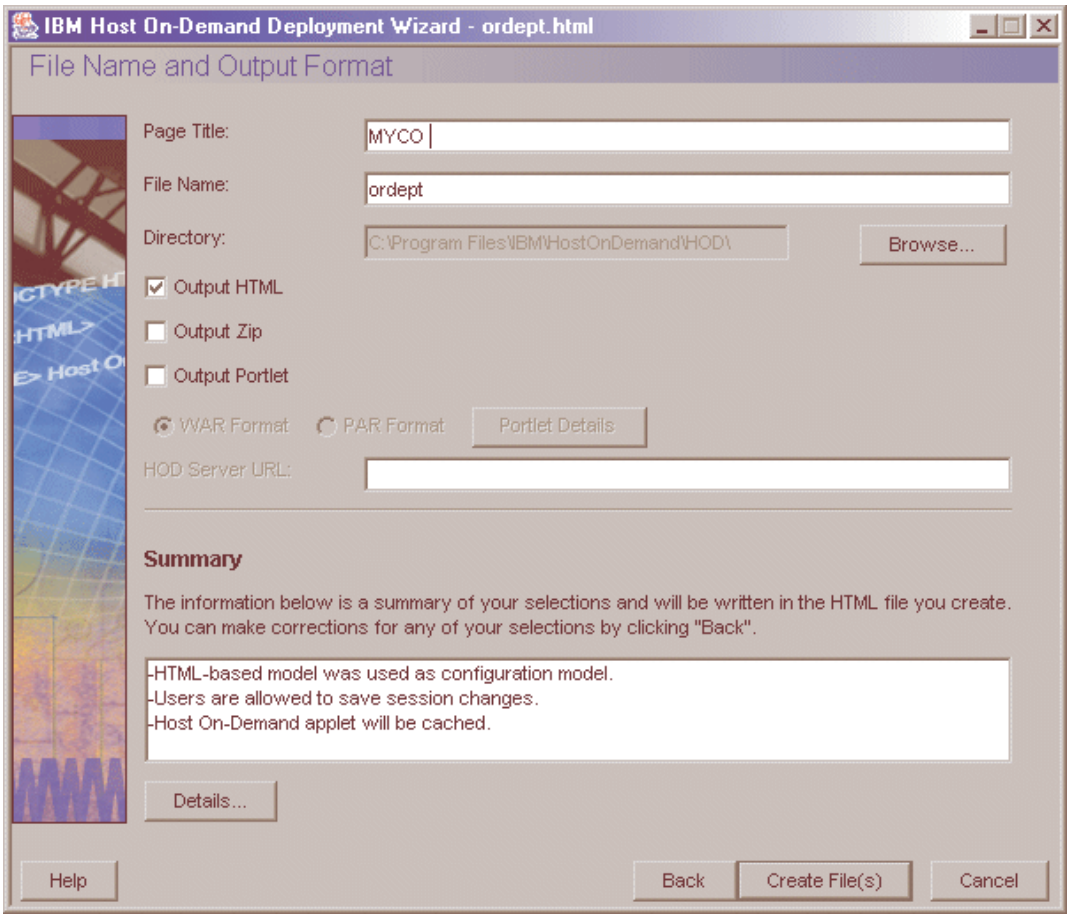
Step 6 of 6: Finish creating your HTML file.

Now that you have configured your Host On-Demand session to use Web Express Logon, you are ready to finish creating your HTML file. At this point, you are still using the Deployment Wizard tool (continued from Step 5).

A. On the Host Sessions window, click Next to open the Additional Options window. Make any changes that you desire and click Next.



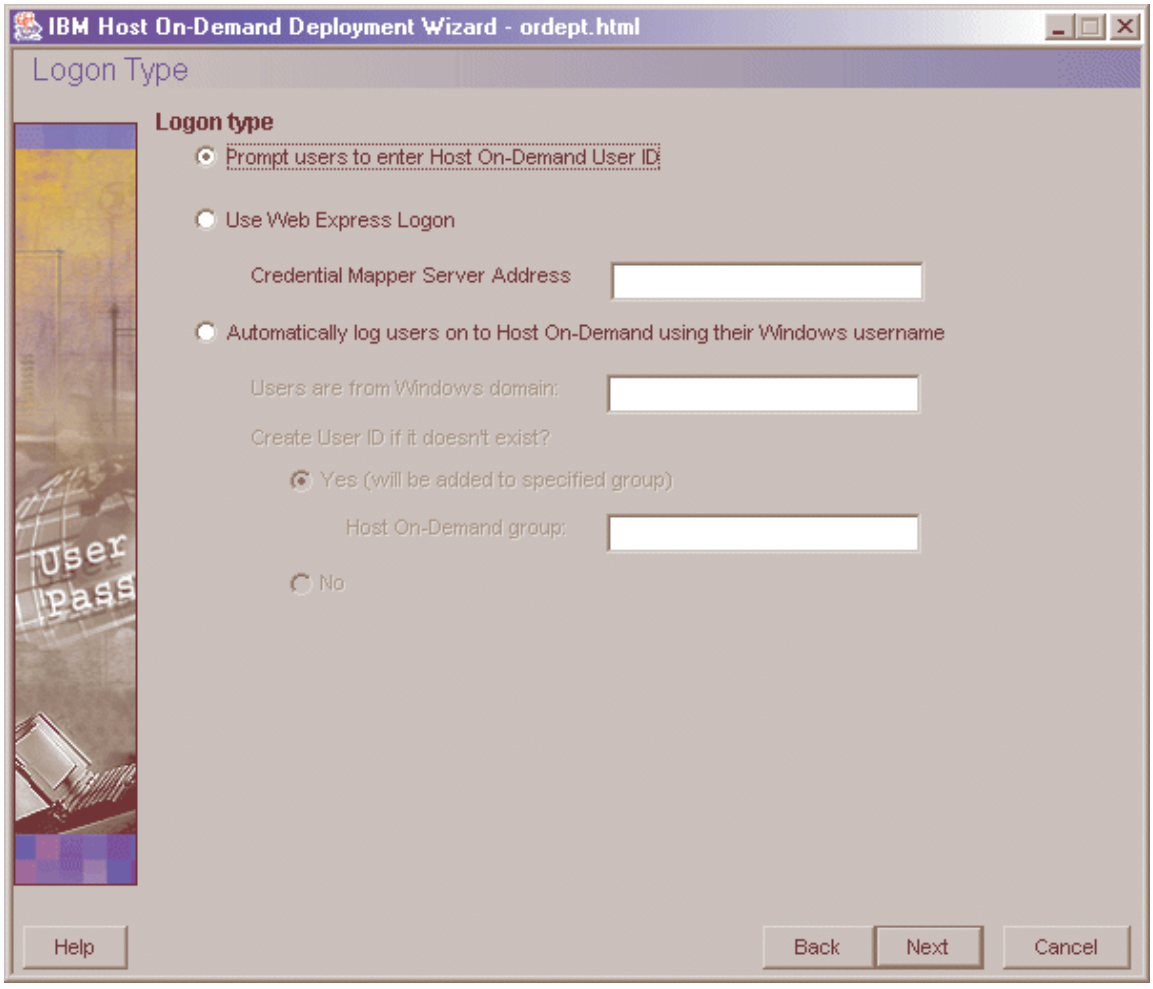
B. On the File Name and Output Format window, enter the page title, the file name, and choose the directory where you want to save your file. You should save it to the Host On-Demand server in a directory known to your Web server; usually, this directory is your Host On-Demand server's publish Directory. Click Create File(s) to finish creating your HTML file.



Congratulations! You have now completed Scenario #3: Connection-based automation: Configuring Web Express Logon in an OS/400 and Kerberos environment. To troubleshoot, refer to Troubleshooting Web Express Logon on page 131.

Web Express Logon using the Configuration server-based model

When creating a HTML file using the Configuration server-based model in the Deployment Wizard, the next window after the Configuration Model window is the Logon Type window.



On this window, you are presented with the following three options:

A. *Prompt users to enter Host On-Demand user ID*: Select this option only if you want users to be challenged for their credentials. This is the default option.

B. *Use Web Express Logon*: Select this option to map the user's network ID to the Host On-Demand ID, which will log users on to the Host On-Demand server.

Note that you must have your user profiles already set up on your Host On-Demand configuration server. If you do not have your user profiles set up and you attempt to launch the HTML file, you will get the following error message:

```
WELM051 User name returned from Web Express Logon is not a
known Host On-Demand user
```

Type the full URL of the credential mapper server, for example, https://*server_name*/*junction*/*cm*/*CredMapper*, where

- *server_name* is the name of the authentication server
- *junction* is the name of the junction point (optional)
- *cm* is the credential mapper servlet space
- *CredMapper* is the servlet name

Selecting this option also requires that you add an additional Vault HCM plug-in and all of its parameters to your web.xml file. For example, take the following steps:

i. Use WebSphere Application Server's Application Assembly Tool to update the following INIT parameter with the new Vault credential mapper name CMPIConfigServer_:

Code example:

```
<init-param>
  <param-name>CMPICredentialMappers</param-name>
  <param-value>CMPIDCASPlugin, CMPIVaultPlugin,
CMPIConfigServer_</param-value>
</init-param>
```

Add the parameter name for the new parameter value specified above, and change the AUTH type to `AuthType_ConfigServer`:

Code example:

```
<init-param>
  <param-name>CMPIConfigServer_</param-name>
  <param-value>com.ibm.eNetwork.security.sso.cms.CMPIVault,
               AuthType_ConfigServer, *</param-value>
</init-param>
```

ii. Prepend the remaining Vault parameters* with the new credential mapper name CMPIConfigServer_, for example, CMPIConfigServer_CMPI_VAULT_DB_ADDRESS. You *do not* need to prepend these two parameters, however:

CMPI_VAULT_DB_HOSTADDR_COL_NAME
CMPI_VAULT_DB_HOSTAPP_COL_NAME

iii. In your Vault HCM database, create a table with the following three columns:

- NETWORKID
- HODID
- PASSWORD

Be sure that the NETWORKID column contains the network IDs, the HODID column contains the Host On-Demand user IDs, and the PASSWORD column contains the Host On-Demand passwords. Since you did not add parameters in your XML file for HOSTADDRESS and APPLICATIONID, you do not need to add the columns for these in your Vault credential database.

C. *Automatically log users on to Host On-Demand using their Windows username:* Select this option to allow Host On-Demand to use the local system's ID for macro-based automation. You can either allow Host On-Demand to use the network ID supplied to the network security application or the Windows system ID to retrieve the host credentials. If you use this option, be sure that you check Use Local Operating System ID in session properties and that you are using

the WAR file that is intended to be used with Windows Domain (wincms.war).

****
*** IMPORTANT**: When using the Configuration server-based model and a network security application such as Tivoli Access Manager, you may be accessing your Host On-Demand pages via a URL such as https://*server_name*/*junction_name*/HOD/myhodpage.html, where *server_name* is the name of the machine running Tivoli Access Manager and *junction_name* is the junction that you create to point to your Host On-Demand server machine and your HTTP server's port number. If this is the case, Host On-Demand will try to contact the Host On-Demand Service Manager to get your user, group, and session information at the server_name rather than at the junction_name.
To remedy this situation, edit the config.properties file found in the HOD directory of your Host On-Demand install directory (\Program Files\IBM\HostOnDemand\HOD\config.properties) by adding this line at the end of the file content:

```
ConfigServer=myhodserver.ibm.com
```

where *myhodserver* is the machine you are pointing to with the *junction_name*.

Customizing Web Express Logon

If you decide to customize Web Express Logon, you may take either of the following two approaches — (1) customize the existing CMS or (2) replace the entire CMS with your own custom version. Although the first approach requires some J2EE knowledge, it is easier to implement than the second approach and does not require experience creating servlets.

The CMS is the core of the credential-mapping framework. It is supplied with Host On-Demand and must be deployed to a J2EE-compliant Web application server. At a high level, the CMS is responsible for the following tasks: (1) determine the client's identity (called a network ID), (2) map the user's network ID to the host ID, and (3) return the host credentials to the client as an XML document. It accomplishes these tasks through credential mapper Java classes called plug-ins. Web Express Logon provides two Network Security plug-ins (one for Tivoli Access Manager and one for Siteminder) to perform the request part of the process and two Host Credential plug-ins (one for DCAS and one for Vault) to perform the response part.

The Network Security plug-in retrieves the user's credentials from the network security application after the user has made an HTTPS request to the CMS. It identifies the user by way of the network user ID and password and then passes it on to the appropriate Host Credential plug-in. The Host Credential plug-in then determines the host user ID and acquires the host access credentials.

If you take the first approach, you can create a Network Security plug-ins and/or a HCM plug-in. For example, if your network security application is not one of three applications supported by Web Express Logon, you can create a Network Security plug-in to meet the requirements of your application. Also, if you want to use an LDAP directory as your HCM database instead of a JDBC database such as IBM DB2, for example, you can customize create your own HCM plug-in.

Approach 1: Replace the entire CMS with your own custom version of the servlet

This document does not describe how to create a servlet, but the following are resources available to help you:

- **IBM Websphere Studio Application Developer**: IBM Websphere Studio Application Developer is the core development environment from IBM. It helps you optimize and simplify J2EE and Web services development by offering best practices, templates, code generation, and the most comprehensive development environment in its class. For more information, refer to http://www.ibm.com/software/awdtools/studioappdev/.
- **IBM developerWorks**: IBM developerWorks is your one-stop developer source. It offers tutorials, training, sample code, CDs and downloads, and more. For more information, refer to http://www.ibm.com/developerworks/ .

If you decide to replace the entire CMS provided with Host On-Demand, you will need to use an HTTP parameter for requests and XML-formatted data for responses. Parameters are supplied to the CMS servlet via an HTTP request, and the response information is encapsulated into an XML-formatted object and returned to the caller.

HTTP request parameters: When Host On-Demand makes a request of the CMS, it applies the appropriate HTTP parameters to this request. This helps determine the needs of the request. Since it must be an HTTP request, the CMS request interface is built around a standard HTTP-style query. Following the HTTPS protocol and server address is the query character, a question mark, and then a list of keys and values. These keys and values are separated by the ampersand symbol. Within each key and value pair, the key and value are separated by the symbol for equality. A sample query may look like the following example:

```
https://www.ibm.com/authserver/servlet/cms?operation=1&destination=www.i
bm.com/somehost&appid=tpf&authtype=AuthType_3270Host
```

The following table is a list of available keys:

Key	Possible value
operation	'1' -- Credential Mapping Request
destination	This is the destination for which the credentials are being requested.
appid	This is the host application ID for which the credentials are being requested.
authtype	This is the type of authentication credentials being requested.
localid	This optional value supplies the user's identification based on the local operating system. For now, this solution is supported only on the Windows operating system.

XML data response object: The CMS returns its response to the client in XML format in an effort to make the response information structured and extensible. This XML format provides a good base for allowing structured access to the return data today and provide for expansion and improvement in the future. The following XML schema defines the format of the XML document:

```
<schema targetNamespace=""
xmlns="http://www.w3.org/2001/XMLSchema">
   <element name="hod-sso-credential" type="hod-sso-credentialType" />
<complexType name="hod-sso-credentialType">
   <sequence>
     <element name="userid" type="string" />
     <element name="password" type="string" />
     <element name="status" type="string" />
   </sequence>
   <attribute name="version" type="string" />
   </complexType>
</schema>
```

Based on the above schema, the following code is a sample of the XML return document that is streamed over the HTTPS connection:

```
<?xml version="1.0"?>
<hod-sso-credential version="1.0" >
   <userid>&^$#^&</userid>
   <password>&^$#^&</password>
   <status>0</status>
</hod-sso-credential>
```

In the above code, the user ID and password elements return garbage characters because they are encrypted. Host On-Demand includes an object called com.ibm.com.eNetwork.HOD.common.PasswordCipher to accomplish this. It contains the following two methods:

public static String encrypt (String plainText)
This method returns an encrypted string passed as a parameter.

public static String decrypt (String cipherText)

This method reverses the encryption process by returning a decrypted string. If the cipherText was not encrypted using the encrypt method, it returns the original input string.

The status element provides the status of the return value. If the credential mapper query fails for any reason, this field reports that failure to the client. Failure codes are defined in the SSOConstants class, which serves as a static repository of related SSO static information. The following table contains the status code definitions:

Status code	Description
0	Success
1	Unknown status code
2	Credential Mapper not found
3	Invalid network user ID
4	Invalid Application ID
5	Invalid server address
6	Database connection error
7	User ID not found in database
8	Exception
9	Invalid user ID
10	Passticket error
11	Timeout
12	Unexpected DCAS return code
13	API not supported
14	Bad URL
15	Unable to parse response
16	Local user ID not available
17	Duplicate XML tags
18	An exception occurred while processing the credential request
19	Network Security plug-in is not defined to the CMS

Approach 2: Customize the existing CMS provided with Host On-Demand

You can create custom Network Security and HCM plug-ins to customize the existing CMS. The CMS relies on these plug-ins to provide the user's network ID and host credentials. The CMS interacts with these plug-ins via the following three Java interfaces, A-C:

A. ***com.ibm.eNetwork.security.SSO.CMS.CMInterface***

The CMInterface interface contains the following methods:

public int Init(Properties p, String id)

This method is used to intialize the plug-in. Any configuration parameters needed to intialize the plug-in will be passed in with the properties object parameter. The parameters are specified in the servlet's web.xml file. The id parameter is the symbolic name of the plug-in specified in the CMS configuration portion of the web.xml file. This value may be used to qualify the instance of the plug-in in the event that multiple instances of the plug-in are running.

public void Destroy()

This method is called when CMS is shutting down.

public CMResponse CMSGetUserCredentials(CMRequest req)

This method is called by the CMS when it has selected the plug-in to respond to a request. If the plug-in is a network security type, it is expected that the plug-in will return the user's

network user id. If the plug-in is a host user credential type, then this method will need to return the user's host credentials.

The following methods are needed for plug-in identification and selection:

public String getName();
This method returns a string that identifes the plug-in.

public String getDescription();
This method returns a string that contains information that describes the purpose and function of the plug-in.

public String getAuthor();
This method is needed to identify the originating company or person of the plug-in.

public String[] getParameters();
This method returns a string array containing the parameter tokens that may be used to configure this plug-in. These tokens are the keys specified in the initialization (INIT) parameters section of the web.xml file used to define the CMS servlet. If no tokens are needed for configuration, the method may return null.

public Properties getParameterInfo(String strParm);
Given a parameter token, this method returns a properties object with the list of properties for the given parameter. The current list of possible properties are as follows:
cmiDefaultValue: This property contains the default value for the specified parameter.
cmiEncrypted: This property determines if the parameter must be encrypted (true or false).
cmiRequired: This property identifies whether or not a parameter is required for initialization of the plug-in.

B. ***com.ibm.eNetwork.security.sso.CMRequest***

The CMRequest object is used by CMS to encapsulate all necessary parameters for a plug-in request. The CMRequest interface contains the following members:

- ID (Host ID or Network ID)
- Host Application ID
- Host Destination Address
- Authentication Type
- HTTP Servlet request object

The CMRequest interface contains the following methods:

public CMRequest()

public CMRequest(String id, String appID, String hostAddr, int authType, HttpServletRequest httpRequest)

public String getID()

public void setID(String id)

public String getHostAppID()

public void setHostAppID(String appID)

public String getHostDestination()

public void setHostDestination(String hostAddr)

public int getAuthType()

public void setAuthType(int authType)

public HttpServletRequest getHttpRequestObject()

public void setHttpRequestObject(HttpServletRequest httpRequest)

public String toString()

C. ***com.ibm.eNetwork.security.sso.CMResponse***
The CMResponse interface encapulates all relevant information needed by the CMS for the request made of a plug-in. The following are its members and methods:

The CMResponse interface contains the following members:

- Status Code
- ID (Host ID or Network ID)
- User Credentials (Password or Passticket)

The CMResponse interface contains the following methods:

public CMResponse()

public CMResponse(Object id, Object password, int status)

public int getStatus()

public void setStatus(int status)

public Object getID()

public String getIDasString()

public void setID(Object id)

public Object getPassword()

public String getPasswordasString()

public void setPassword(Object password)

public String toString()

Writing your own plug-ins

The Network Security and HCM plug-ins are Java classes that implement the CMInterface interface. The CMS makes calls to your plug-ins via the APIs described earlier.

Network Security plug-in: Host On-Demand provides two Network Security plug-ins, one for Tivoli Access Manager and one for Netegrity Siteminder. If you decide not to use either of these, you may create your own plug-in.

The primary function of the Network Security plug-in is to acquire the user's network ID, which may be gleaned from the HTTP header of the incoming HTTP request object. The details of how to acquire the network ID is specific to your network security application. Refer to your network security documentation for more information.

HCM plug-in: Host On-Demand provides two Host Credential plug-ins, one for DCAS and one for Vault. If you decide not to use either of these, you may create your own plug-in.

The primary function of the HCM plug-in is to take the user's network ID (and perhaps the application ID) and obtain the appropriate host credentials. In Web Express Logon's implementation, users' network IDs are mapped to their host IDs by way of a JDBC-accessible database. However, you may wish to do this by another means, such as LDAP. For this reason, you may want to write your own HCM plug-in. In our DCAS/JDBC plug-in, we automate 3270 application logins by associating users' network IDs to their host IDs. Then, the host IDs and application IDs are used to obtain a RACF-generated passticket. This passticket is then used to sign the user on to the host. In your environment, you may not want to use the JDBC association aspect of our plug-in. For this reason, we have provided a DCAS API that you can use to develop your own custom plug-ins. This API provides access to RACF-generated passtickets.

The DCAS API object (DCASClient) encapsulates the Passticket requests:

The DCAS API client has the folllowing members:

- Port Number
- Keyring File Name
- Keyring Password
- Use WellKnownTrustedCAs
- Server Authentication
- Trace Level
- Trace Log File Name

The DCAS API client has the folllowing methods:

Public DCASClient()

This constructor should be used if you want to use the default trace level and log file name when the object is created.

Public DCASClient(int traceLevel, String logFile)

- traceLevel - Trace level (0=None, 1=Minimum, 2=Normal and 3=Maximum)
- logFile - Trace log file name. It should include the full path name.

This constructor should be used if you want to specify a trace level and log file name when the object is created.

Public int Init(int dcasPort, String keyringFileName, String keyringPassword)

- dcasPort - DCAS server's port number. If not specified, the default port number of 8990 will be used.
- keyringFileName - The name of the SSL keyring database file. It should include the full path name.

- keyringPassword - The password of the above keyring database.

This method should be called after creating the DCASClient object. The parameters are stored in the object, and they do not change for the life of the object. The keyringFileName should include the full path name. The keyring database must contain DCAS client certificate. It should also contain the DCAS server certificate if it is self signed or from an unknown Certificate Authority. The keyring Password should have been encrypted using the encrypt password tool. It will be decrypted before being stored in the object. The valid return codes are described in the SSOConstants object.

Public void setWellKnownTrustedCAs(boolean wellKnownCAs)

Public void setServerAuthentication(boolean serverAuth)

Public CMResponse getPassticket(Sting hostUserID, String hostApplID, String hostAddr, long timeout)

- hostUserID - User ID for which the passticket is being requested.
- hostApplID - Application ID for which the passticket is being requested.
- hostAddr - The DCAS server's address.
- timeout - The time available for the DCAS protocol to return a passticket. It is specified in milliseconds.

This method should be called after creating and initializing the DCASClient object to obtain a passticket from the DCAS server. The passticket and the user ID are returned in a CMResponse object. The caller should check the status field of the CMResponse object to see if the call was successful or not. If the call was successful, the status field will be set to SSO_CMR_SUCCESS. The valid values for the status field are specified in the SSOConstants object. An SSL client authenticated connection is established with the DCAS server, and it is reused for all subsequent passticket requests.

Public void Destroy()

This method closes the DCAS connection.

Troubleshooting Web Express Logon

Web Express Logon depends on a number of independent processes working together to function properly. Some of these processes are client-based while others are host-based. If one or more of these processes break down, you must be able to determine which process is causing the problem in order to resolve it appropriately.

If you have problems with Web Express Logon, analyze the type of results you receive and any accompanying informational messages. Some of these informational messages are included as part of the Host On-Demand client by way of an interactive panel, and/or they may be part of a server-based log.

If Web Express Logon is not functioning properly (that is, you are not logged in a host emulation session), complete the following checklist to try to determine the root cause:

Checklist:

A. Did the Host On-Demand client display an error message?

- If yes, skip to Web Express Logon client-side messages.
- If no, verify the following on your session configuration panel:

B. Did you see the **NVT5250 : IBMTICKET getPassticket failed with error x** error message in the Java Console?

This error tells you that Host On-Demand is not successfully acquiring a passticket and there is a problem in your setup. If you do not see this error, Host On-Demand *is* successfully getting a passticket, but there are problems with the host accepting the passticket.

C. Have you enabled Express Logon for the session that you are currently running? To do this, highlight your session and select Properties under the Configure drop-down menu in the Deployment Wizard. On the left side of the window, select Express Logon under Connection and click Yes to enable Express Logon.

D. Is this a 5250 session and you are using a Kerberos passticket for authentication? If so, you will need to make sure you select Yes for the Use Kerberos Passticket option on the Express logon window of session properties.

E. Are you using macro-based automation? If so, verify the following items:

- When creating the macro, verify that you selected Web Express Logon (not Certificate Express Logon) on the Record macro window.
- If you are expecting the macro to run when the session is started, verify that you have selected Auto-Start macro in your session configuration.

F. Did your automation macro run but not provide the appropriate credentials to log in the user? This means that you have properly accessed the Credential Mapper Web application, but something is not functioning properly within that environment. You should enable server-side logging and attempt another credential automation event. Then look in the log that is created and refer to Web Express Logon server-side messages.

G. Are you using IBM WebSphere Application Server and have Java 2 security enabled? If so, check to make sure that the following permissions are granted in the was.policy file, which is located in the META-INF directory.

```
permission java.io.FilePermission "<<ALL FILES>>", "write";
```

You can change <<ALL FILES>> to whichever directory you specifed in the CMPI_TRACE_LOG_FILE parameter in the web.xml file.

```
permission java.lang.RuntimePermission
"accessClassInPackage.sun.jdbc.odbc";
```

This applies to the JDBC database Host Credential Mapper (HCM).

Web Express Logon client-side messages: When an unexpected problem occurs during the Web Express Logon process, the Host On-Demand client provides information about the problem to the user by displaying a panel with an informational message. Each of these messages contains an error code that you can use as a unique identifier for the problem that is occurring.

The following is a list of all Web Express Logon messages for the Host On-Demand client.

WELM001: Message key not found: status = value
This message should only be seen in the event of an error found in a custom plug-in. If you have customized the Web Express Logon credential mapper framework, you can create user defined error codes. If the Web Express Logon credential mapper returns such a code, this message will be displayed.

WELM002: No suitable Host credential plug-in found
This message is displayed when there is no appropriate credential plug-in found to handle the Host On-Demand client's credential request. Verify that your Web Express Logon credential mapper application is properly configured to handle the Host On-Demand client's session type.

WELM003: Invalid network user ID
The Web Express Logon credential mapper cannot acquire the user's network ID. This can be caused by improper settings in the network security plug-in section of the CMS configuration. If the local operating system identification is being used to identify the user, make sure this option is selected in the Express Logon section of the Session Configuration panel.

WELM004: Invalid Application ID
This message indicates the lack of a valid Application ID. You specify the Application ID when you create the Web Express Logon macro. When you create the macro, be sure that you enter the proper value for the Application ID.

WELM005: Invalid server address
This message indicates the lack of a valid server address. The server address is specified as the Destination Address on the Session Configuration panel. For some credential plug-ins, this is a required parameter.

WELM006: Could not connect to database
This problem can be generated by an improperly configured database link. Please verify that the database is properly configured in your CMS configuration. If the configuration information looks correct, you should independently verify the database's availability and running status. The database's configuration and management tools are a good place to perform this test.

WELM007: A matching user ID not found in database
The credential plug-in is not able to find a match for the user's host ID, given the search criteria. Verify that the user's host ID is specified in the database or other storage medium used by the credential plug-in. In addition, you may want to enable server-side logging and verify that the parameters being sent to the CMS are correct.

WELM008: The Credential Mapper Servlet reported an exception while processing a credential request. Please see the server log for details.
This generalized message is a result of an exception occurring on the CMS. Please follow the instructions for enabling server-side logging for more information about the cause of this problem.

WELM009: Invalid User ID
A credential plug-in does not have a valid user's host ID. For some plug-ins, the host ID is used to obtain a temporary passticket credential to access the host. If the value used is not appropriate, this message is generated. You may want to verify the user's host ID is specified

in the database or other storage medium used by the credential plug-in. In addition, you may want to enable server-side logging and verify that the parameters being sent to the CMS are correct.

WELM010: Passticket could not be obtained

This message is displayed when a credential plug-in receives an error during the passticket creation process. Typically, the actual creation of the passticket occurs in a process outside of the credential plug-in. If that external process returns an error, this message displays. You should enable server-side logging and perform the credential request again. Using the information in the log along with the messages found in this section of the document should provide a better understanding of the problem.

WELM011: Credential/Passticket request timed out

This message is the result of a pending request timing out before it could be resolved. This could happen when the Host On-Demand client is making a request of the Credential Mapper Server, or it could be the credential plug-in making a request of an external entity. In either case, if the default time elapses before the request is fulfilled, this message is generated. To rectify the problem, verify that the addresses being used are correct. For the Host On-Demand client, the Credential Mapper server is specified as the Credential Mapper Server address in the Express Logon properties window of the Session Configuration panel. If the credential plug-in is generating this problem, verify that the credential plug-in is properly configured in your CMS configuration.

WELM012: Unexpected return code received from DCAS

This error is created when a credential plug-in receives an unexpected return value of an external application. You should enable server-side logging and perform the credential request again. Using the information in the log along with the messages found in this section of the document should provide a better understanding of the problem.

WELM013: API not supported. Contact the system administrator for server log.

This message informs the user that an unsupported request has been made of the credential plug-in selected by the credential mapping application. You should enable server-side logging and perform the credential request again. Using the information in the log along with the messages found in this section of the document should provide a better understanding of the problem.

WELM014: A malformed URL was specified for the Credential Mapper Server Address

The address used for the Credential Mapper server is not a valid URL address. The Credential Mapper server is specified as the Credential Mapper server address in the Express Logon properties of the Session Configuration panel.

WELM015: Unable to parse Credential Mapper response

The response generated by the Credential Mapper server application contains a response that is improperly formatted. This may happen when a custom Credential Mapper server application is used in place of the default Host On-Demand Credential Mapper server application. Refer to Customizing Web Express Logon on page 124 for more information about the CMS response format.

WELM016: Local user ID not available

This message is generated when the operating system on which the Host On-Demand client is running does not support the Use Local Operating System ID option for network security identification. Refer to the Introduction for more information about which operating systems and versions are supported by this option.

WELM017: Credential Mapper response contained a duplicate userid, password, or status tag

This problem is caused when the response generated by the Credential Mapper server application contains duplicate response values. This may happen when a custom Credential Mapper server application is used in place of the default Host On-Demand Credential Mapper server application. Refer to Customizing Web Express Logon on page 124 for more information about the CMS response format.

WELM018: An exception occurred while processing the credential request: some exception

This message is displayed when an exception occurs in the Host On-Demand client during the Web Express Logon process. If the exception is an IOException, the problem may be the Credential Mapper server address specified in the Express Logon properties panel in the session configuration. If the address seems correct, validate that the CMS server is available. Typing the Credential Mapper address specified in the session configuration into the address entry field of your browser allows you to test access to the CMS server easily. The results should be an XML document similar to the one described earlier in this document.

WELM050: Web Express Logon Credential Mapper Server Address not specified

Web Express Logon is used to automate the Host On-Demand configuration server login process, but the Credential Mapper server address is not specified. Verify that you have specified the proper value for the Credential Mapper server address in the Deployment Wizard.

WELM051: User name returned from Web Express Logon is not a known Host On-Demand user

Web Express Logon is used to automate the Host On-Demand configuration server login process and the user name provided by Web Express Logon is not a valid Host On-Demand user. Verify that the user is listed in the Host On-Demand configuration by accessing the Host On-Demand Administrative Console. In addition, view the server-side log to verify that the user name is being retrieved properly.

WELM052: Invalid password returned from Web Express Logon

Web Express Logon is used to automate the Host On-Demand configuration server login process, and the password provided by Web Express Logon is not a valid. Verify that the user is listed in the Host On-Demand configuration by accessing the Host On-Demand Administrative Console. In addition, view the server-side log to verify that the user name is being retrieved properly.

WELM053: This session is not enabled for Web Express Logon

A Web Express Logon macro is executed, and the session on which it is running has not been configured to use Web Express Logon. Web Express Logon can be configured via the Host On-Demand session configuration panel.

Web Express Logon server-side messages: The following are the primary server-side messages:

CMPIE001: Credential Mapper Plug-in initialization failed for: YourCredentialMapperName
This error occurs when the Credential Mapper plug-in corresponding to YourCredentialMapperName fails to initialize successfully. Possible causes of this error include the following:

- Your web.xml specifies an invalid or missing value for a parameter that is required by the specified plug-in.
- To determine which parameter(s) is causing the problem, turn on tracing for the plug-in and look in the log for error CMPIE008.
- You are using the DCAS or Vault plug-ins, and an error occurs when attempting to connect to the credentials database. Turn on tracing for the plug-in to obtain more diagnostic information (database driver missing, SQL exception, etc).
- You are using a custom plug-in, and your Init() method is returning a value other than 0 on success. Refer to the Customizing Web Express Logon on page 124 for more information about writing your own HCM plug-in.
- You are using DCAS, and the SSL key database file or password is not specified in web.xml.

CMPIE003: No CM configuration can be found for the CM identified by the YourCredentialMapperName name.
This error occurs as a result of a missing element in your web.xml file. If you provide a value for the CMPICredentialMappers parameter that is not also a parameter itself elsewhere in the web.xml, you will get this error. For example, if you have the following definition in your web.xml,

```
<init-param>
<param-name>CMPICredentialMappers</param-name>
<param-value>vault</param-value>
</init-param>
```

you would also need something like this,

```
<init-param>
<param-name>vault</param-name>
<param-value>com.ibm.eNetwork.security.sso.cms.CMPIVault,AuthType_327
0Host,*</param-value>
</init-param>
```

or you would get the error above.

CMPIE004: No Credential Mappers have been specified.
This error occurs when your web.xml does not define the CMPICredentialMappers parameter. Be sure to include the following in your web.xml:

```
<init-param>
<param-name>CMPICredentialMappers</param-name>
<param-value>YourCredentialMapperName(s)</param-value>
</init-param>
```

CMPIE005: No Credential Mapper found for Auth type: AuthTypeValue
When you define a Credential Mapper in your web.xml, you specify the type of Authentication to which the plug-in applies. For example, if you had an entry such as the following,

```
<init-param>
```

```
<param-name>vault</param-name>
<param-value>com.ibm.eNetwork.security.sso.cms.CMPIVault,AuthType_327
0Host,*</param-value>
</init-param>
```

this would show that the vault Credential Mapper is only intended to be used with 3270 host sessions. If this were the only Credential Mapper defined in your web.xml and you tried to perform a logon to a 5250 session, you would receive this error with AuthTypeValue equal to AuthType_5250Host. Be sure that your web.xml has a Credential Mapper defined that is appropriate for your authentication type.

CMPIE007: No authentication type specified for CM object: YourCredentialMapperName
When you define a Credential Mapper in your web.xml, you must specify the full class path name, the authentication type, and the host mask. If you do not specify an authentication type, or if you specify an invalid authentication type (such as AuthType_Fred), you will get this error. For a list of valid authentication types, refer to the glossary.

CMPIE008: Invalid value for parameter: ParameterName
This error occurs when a parameter that is required by the plug-in has an invalid value or has not been specified. Provide an appropriate value in the web.xml for the parameter ParameterName.

CMPIE010: Exception and Host User ID not found for Network ID: NetIDValue.
An exception occurred before the host user ID corresponding to NetIDValue could be found. A possible cause of the exception is a mismatch between the column names in the data source and the column names specified in the web.xml. Another possibility is an error in the formatting of the table name ([tableName$] for Excel, simply tableName for DB2). Double check your web.xml for errors and refer to the exception trace in the server log for debugging information.

CMPIE011: Host User ID not found for Network ID: NetIDValue.
This error occurs when there is no entry found in the database for NetIDValue. Check your database and verify that there is an entry for NetIDValue. Make sure that the host address and application ID found in the server log for this query match the host address and application ID specified for this NetID in the database.

CMPIE012: SQLException: Value.
This error occurs when attempting to open or close a connection to the database. Make sure that the database is available and correctly specified in the web.xml file.

CMPIE013: Exception: Value.
An exception occurred in the plug-in code. The cause of the exception is indicated in the Value message.

DCAS error messages: The following are the primary DCAS error messages:

DCASE001: Cannot import the CA certificates contained in Keyring Database.
An SSL runtime exception occurred while loading the CA certificates from the KeyringDatabase. The file may be corrupted. Please see the additional logged messages for details. You may have to set the CMPI_DCAS_TRACE_LEVEL parameter in web.xml to 3 to see the additional messages.

DCASE002: Cannot read the keyring file: KeyringFileName
The specified KeyringFileName cannot be loaded. Make sure that the file exists and the path name and file name are correctly specified in the web.xml file. See the exception trace for additional information.

DCASE003: The DCAS server address is either blank or null.
The Host On-Demand client's credential request contains an invalid server address. See the WELM005 message for details.

DCASE004: The Keyring file name is either blank or null.
The CMPI_DCAS_KEYRING_FILE parameter must be specified in the web,xml file. Check the web.xml file.

DCASE005: The Keyring password is either blank or null.
The CMPI_DCAS_KEYRING_PASSWORD parameter must be specified in the web.xml file. Check the web.xml file.

DCASE006: The host user id is either blank or null.
The host user ID retrieved from the vault database is either blank or null. Check the vault database for host user ID.

DCASE007: The host application id is either blank or null.
The Host On-Demand client's credential request contains an invalid application ID. See the WELM004 message for details.

DCASE008: Passticket could not be obtained for user ID: Userid
The DCAS client could not obtain a passticket for the specified User ID. Make sure that the host user ID is valid and it is defined to the host credential system such as RACF. Also, see the additional logged message for a specific failure.

DCASE009: DCAS timer expired - no response from server: Host
The DCAS connection timer expired before a passticket request could be completed. If this problem persists, you may want to increase the value of the CMPI_DCAS_REQUEST_TIMEOUT parameter in the web.xml file. This value should be less than the timeout value for the macro.

DCASE010: Unexpected DCAS return code: ReturnCode
This error suggests an internal coding error. Please make a note of the ReturnCode and report this problem.

DCASE013: DCAS Exception: Exception
Exception occurred while processing a passticket request. See the additional logged messages for details.

DCASE021: Cannot send passticket request to server Host
The DCAS server connection is not active. Check the DCAS server log and retry the operation.

DCASE022: An unexpected error occurred while processing a passticket request.
An unexpected exception occurred while processing a passticket request. See the exception details to determine the cause of the problem.

DCASE023: An error occurred while receiving data from the passticket server Host. The connection is closing.
Input/Output error occurred while receiving data from the passticket/DCAS server. Retry the operation. If the problem persists, check the DCAS server log for details.

DCASE050: Cannot create socket to the passticket server at IpAddr. See other messages for details.
An SSL exception occurred while creating a secure connection. See the additional logged messages for details. You may have to set the CMPI_DCAS_TRACE_LEVEL parameter in web.xml to 3 to see the additional messages. This message typically indicates an SSL handshake failure.

DCASE051: The DCAS server at Ipaddr is an unknown host.
The Destination Address specified in the Session Connection panel is an unknown host. Check the Ipaddr to make sure it is valid. See the WELM005 message.

DCASE052: Cannot create socket to the passticket server at IpAddr because of an I/O error.
An I/O exception occurred while creating a secure connection to IpAddr. See the additional logged messages for details. You may have to set the CMPI_DCAS_TRACE_LEVEL parameter in web.xml to 3 to see the additional messages. The server at IpAddr may be down.

DCASE060: The common name in the certificate received from Host is empty. SSL connection is terminated.
The SSL server authentication failed. The Host presented a certificate that does not contain the common name. Please update the server certificate's common name, or turn the server authentication off.

DCASE061: The common name in the certificate received from Host has no address. SSL connection is terminated.
The SSL server authentication failed. The host presented a certificate whose common name does not have an address. Update the server certificate's common name to the server's IP address, or turn the server authentication off.

DCASE062: The passticket server's name Host has no address. SSL connection is terminated.
The SSL server authentication failed. The host presented a certificate whose host name does not have an IP address. Make sure that an IP address is associated with the host name, or turn the server authentication off.

DCASE063: The common name in the certificate received from Host does not match the partner's common name. SSL connection is terminated.
The SSL server authentication failed. The socket or discovered address does not match the common name specified in the Host certificate. The server certificate could not be authenticated. Update the server certificate's common name to match its IP address, or turn the server authentication off.

DCASE064: No certificate chain received from Host. SSL connection is terminated. The host did not present its certificate when a connection was established. The server certificate could not be authenticated. The host must be configured to send its certificate to do the server authentication.

Password Encyption Tool

Host On-Demand provides a password encryption tool so you can encrypt your passwords for added security. It is a command-line tool that allows you to generate a file that stores the encrypted password, which you must then copy to the appropriate place in the web.xml file. The HCM plug-in decrypts the password before using it.

If you create a custom Host Credential plug-in, the plug-in should use the com.ibm.eNetwork.HOD.common.PasswordCipher object to decrypt the password. The CLASS file for this object is included in WAR file. Refer to Custom Credential Mapper Servlet response object for a description of the encrypt and decrypt methods.

Windows platforms: Using a DOS prompt, change the current directory to the Host On-Demand's bin directory and type the following command:

```
encrypt <password> [filename]
```

where *password* is the password to be encrypted and *filename* is the name of the file that you want to use to store the encrypted password. The default filename is password.txt.

Unix platforms: Issue the following command:

```
cd your_install_dir
Java -classpath .;your_install_dir\lib\sm.zip \
com.ibm.eNetwork.security.sso.cms.tools.Encrypt <password> [filename]
```

where *your_install_dir* is your Host On-Demand installation directory, *password* is the password to be encrypted, and *filename* is the name of the file that you want to store the encrypted password. The default filename is password.txt.

Glossary of terms

Authentication type
This parameter value is used to identify the type of authentication that the requestor needs. Once you specify the desired authentication type, the CMS can better identify which HCM plug-in to select to handle the request. You can pair multiple authentication types together to give HCM plug-ins the freedom to support multiple authentication types. Use the vertical bar character to join multiple authentication types. The five identified authentication types and descriptions are listed in the following table:

Authentication type	Description
AuthType_3270Host	Identifies the credentials to be used with a 3270 emulation
AuthType_5250Host	Identifies the credentials to be used with 5250 emulation
AuthType_VTHost	Identifies the credentials to be used with VT emulation
AuthType_FTPPassword	Credentials used to access an FTP host
AuthType_ConfigServer	Credentials identified by the token used to identify the user to the Host On-Demand configuration server (if you are using the Configuration server-based model)
AuthType_All	Identifies the credentials to be used with all authentication types

Credential Mapper Servlet (CMS)
The CMS is the core of the credential-mapping framework. It is supplied with Host On-Demand and must be deployed to a J2EE-compliant Web application server. At a high level, the CMS is responsible for the following tasks: (1) determine the client's identity (called a network ID), (2) map the user's network ID to the host ID, and (3) return the host credentials to the client as an XML document.

Digital Certificate Access Server (DCAS)
DCAS is a TCP/IP server that runs on z/OS and OS/390 platforms. DCAS clients connect to DCAS using Secure Socket Layer (SSL). The purpose of DCAS is to receive an application ID and a host ID from a DCAS client and then ask RACF to return a valid passticket for the input user ID and application ID.

Enterprise Identity Mapping (EIM)
EIM is designed to help you manage multiple user registries and user identities in your enterprise. EIM is a mechanism for mapping (associating) a person or entity to the appropriate user identities in various registries throughout the enterprise. EIM provides APIs for creating and managing these identity mapping relationships, as well as APIs that applications use to query this information. iSeries Navigator, the iSeries graphical user interface, provides wizards to configure and manage EIM. In addition, administrators can manage EIM relationships for user profiles through iSeries Navigator. The iSeries server uses EIM to enable OS/400 interfaces to authenticate users by means of network authentication service. Applications, as well as OS/400, can accept Kerberos tickets and use EIM to find the user profile that represents the same person as the Kerberos ticket represents.

Full class path name
The CMS uses the value of the full class path name to create a class object of the specified type. That object is then used to handle CMS or HCM plug-in requests. The specified class file must be in the ...\WEB-INF\classes subdirectory in a loose file (not as a JAR file). From this location, the CMS will be able to access and use it whenever the need arises.

HCM database
The HCM is a back-end repository that maps users' network IDs to their host IDs. This repository can be a JDBC database such as IBM DB2. The DCAS and Vault plug-ins provided with Web Express Logon are designed to work with a such a database. Another possibility for a repository is

an LDAP directory. However, using LDAP as your HCM database requires you to write your own plug-in.

HCM plug-in
Host On-Demand provides two Host Credential plug-ins, one for DCAS and one for Vault. The primary function of the HCM plug-in is to take the user's network ID (and perhaps the application ID) and obtain the appropriate host credentials. In Web Express Logon's implementation, users' network IDs are mapped to their host IDs by way of a JDBC-accessible database.

Host ID
A host ID is the credential used to uniquely identify the user to the host being accessed.

Host mask
The host mask is a secondary selection criteria used by the CMS to identify the most appropriate HCM plug-in. This value can contain one or more host addresses. Use the vertical bar character to join multiple addresses. Use the asterisks character to wildcard a host address. The wildcard character may start, end, or start and end a host address. The following table lists valid wild-carded addresses:

Host mask	Value matched
*.raleigh.ibm.com	Matches all addresses that end with .raleigh.ibm.com
ralvm*	Matches all addresses that start with ralvm
*	Matches all
xyz	Matches any host address that contains xyz

Junction (WebSEAL)
A junction is a TCP/IP connection between the front-end WebSEAL server and the destination host.A junction allows WebSEAL to provide protective services on behalf of the back-end server. WebSEAL can perform authentication and authorization checks on all requests before passing those requests on to the back-end server. Junctions between cooperating servers result in a single, unified, distributed Web space that is seamless and transparent to users. The client never needs to know the physical location of a Web resource. WebSEAL translates logical URL addresses into the physical addresses that a back-end server expects.

Kerberos
Kerberos is an authentication protocol that grants users access to individual services in a computer network. Users request tickets from the KDC, and if authenticated, they use a Kerberos ticket to access resources within the network without their passwords being sent through the network.

Key Distribution Center (KDC)
A KDC is a network service that provides tickets and temporary session keys. The KDC maintains a database of principal names (users and services) and their associated secret keys. It is composed of the authentication server and the ticket granting ticket server. You should always use a secure machine to act as your KDC because if an unauthorized user gained access to the KDC, your entire realm could be compromised.

Network authentication service (NAS)
NAS allows the iSeries server and several iSeries services, such as iSeries Access for Windows, to use a Kerberos ticket as an optional replacement for a user name and password for authenticating a user. Kerberos protocol, developed by Massachusetts Institute of Technology, allows a principal (a user or service) to prove its identity to another service within an insecure network. Authentication of principals is completed through a centralized server called a key distribution center (KDC). The KDC authenticates a user with a Kerberos ticket. These tickets prove the principal’s identity to other services in a network. After a principal is authenticated by these tickets, they can exchange encrypted data with a target service. NAS verifies the identity of

a user or service in a network. Applications can securely authenticate a user and securely pass on his or her identity to other services on the network. Once a user is known, separate functions are needed to verify the user's authorization to use the network resources.

Network ID
A network ID is the credential that uniquely identifies the user to the network security application. The CMS calls upon the Network Security plug-in to acquire the user's network ID from the network security application.

Network Security plug-in
Host On-Demand provides two Network Security plug-ins, one for Tivoli Access Manager and one for Netegrity Siteminder. The primary function of the Network Security plug-in is to acquire the user's network ID, which may be gleaned from the HTTP header of the incoming HTTP request object. The details of how to acquire the network ID is specific to your network security application.

Passticket
A passticket is a credential that is similar to a password, however a passticket expires after a certain period of time and is used only one time.

Principal name
The name of a user or service in a Kerberos network. A user is considered to be a person where a service is used to identify a specific application or set of operating system services. On iSeries, the krbsvr400 service principal is used to identify the service used by iSeries Access for Windows, QFileSrv.400 and Telnet servers when authenticating from the client to the iSeries.

Resource Access Control Facility (RACF)
RACF is an IBM security product that protects resources by granting access to only authorized users of protected resources. RACF retains information about the users, resources, and access authorities in profiles on the RACF database and refers to the profiles when deciding which users should be permitted access to protected system resources.

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.

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.
For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan
The following paragraph does not apply to the United Kingdom or any other country or region 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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department T01
Building B062
P.O. Box 12195
Research Triangle Park, NC 27709-2195
U.S.A.
Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee. The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

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

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

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation.

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