IBM WebSphere Host On-Demand Version 8.0

Web Express Logon Reference

SC31-6377-00

IBM WebSphere Host On-Demand Version 8.0

Web Express Logon Reference

SC31-6377-00

Note

Before using this information and the product it supports, read the information in Appendix D, “Notices,” on page 79.

First Edition (September 2003)

This edition applies to Version 8.0 of IBM® WebSphere Host On-Demand (program number 5724–F69) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Part 1. About this book

Chapter 1. Description of book

This book describes a step-by-step approach to understanding, planning for, implementing, and troubleshooting Web Express Logon. It is written for administrators who are interested in learning about and implementing Web Express Logon in their computing environment. It contains eight parts:

- Overview of Web Express Logon
- Planning
- Implementation
- Real-life scenario
- API programming guide
- Troubleshooting error messages
- Appendices
  - Password encryption tool
  - Glossary of terms
  - Sources for more information
  - Notices
  - Trademarks

The *Host On-Demand Web Express Logon Reference* is also available in PDF format on the Host On-Demand Version 8 CD-ROM and in the Host On-Demand PDF library at http://www.ibm.com/software/webservers/hostondemand/library/v8infocenter/.

Conventions used in this book

The following typographic conventions are used in *Host On-Demand Web Express Logon Reference*:

Table 1. Conventions used in this book

Convention	Meaning
`Monospace`	Indicates text you must enter at a command prompt and values you must use literally, such as commands, functions, and resource definition attributes and their values. Monospace also indicates screen text and code examples.
Italics	Indicates variable values you must provide (for example, you supply the name of a file for *file_name*). Italics also indicates emphasis and the titles of books.
>	When used to describe a menu, shows a series of menu selections. For example, “Click File > New” means “From the File menu, click the New command.”
	When used to describe a tree view, shows a series of folder or object expansions. For example, “Expand HODConfig Servlet > Sysplexes > Plex1 > J2EE Servers > BBOARS2” means: 1. Expand the HODConfig Servlet folder 2. Expand the Sysplexes folder 3. Expand the Plex1 folder 4. Expand the J2EE Servers folder 5. Expand the BBOARS2 folder

✏	This graphic is used to highlight notes to the reader.
💡	This graphic is used to highlight tips for the reader.
▦	This graphic refers to connection-based automation for IBM eServer iSeries environments that support Kerberos authentication.
🔧	This graphic is used to indicate troubleshooting tips for the reader.

Part 2. Overview of Web Express Logon

Chapter 2. Introduction

In the age of e-business on demand, finding ways to simplify the user experience while maintaining company security can be a real challenge. For example, many companies would like to decrease the number of IDs and passwords that their users have to manage, but they also realize that allowing users to access company resources without proper identification risks company security.

Several products exist in the marketplace that claim to solve the multiple logon issue and maintain security at the same time. However, these products generally apply to Web-based applications only and do not address logon processes for legacy hosts and host-based applications. In other words, in host-based applications that do not use HTML or XML, automating the logon process requires being able to intercept the telnet data stream. Because of its unique position to work with individual screens and the ability to substitute fields in the data stream, Host On-Demand is an ideal candidate to address multiple logon issues in companies whose users access host systems via browser-based terminal emulation.

Web Express Logon works in conjunction with your company's network security application to maintain company security while allowing users to log on to host systems without having to re-enter their user IDs and passwords. It has several benefits, including 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 (ELF)?

Host On-Demand offers two types of Express Logon:

- Web Express Logon
- Certificate Express Logon

Web Express Logon is a new feature available with Host On-Demand Version 8. Certificate Express Logon, formerly known as Express Logon Feature (ELF), has been available since Host On-Demand Version 5. Although the name has changed, Certificate Express Logon functions the same as ELF did in earlier versions and requires the same configuration.

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 z/OS and OS/390 host systems. 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 SSL configuration nor client-side certificates, and it can function on multiple platforms. 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.

How does Web Express Logon work?

The overall goal of Web Express Logon is to provide an automated way for users to log on to hosts and host-based applications without having to provide an additional ID. It is designed to function within a wide range of computing environments. Your particular environment determines the way in which you plan for, implement, and use Web Express Logon.

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 type. Macro-based automation is for environments of varying host types that *are not* using Kerberos authentication. As the name implies, it requires you to create a macro to perform logon automation. Connection-based automation, on the other hand, is *only* for IBM eServer iSeries host environments that support Kerberos authentication. It does not require a macro to perform logon automation but instead relies on a telnet feature to supply the user's necessary logon information.

The following sections provide more details about Web Express Logon's macro-based and connection-based automation:

Macro-based automation

In order to use the macro-based automation style of Web Express Logon, you must have a network security application in place. Host On-Demand provides out-of-the-box support for three common network security applications without requiring additional coding: IBM Tivoli Access Manager, Netegrity Siteminder, and 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 Chapter 7, "Customizing Web Express Logon," on page 53.

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)

The CMS is supplied with Host On-Demand and must be deployed to a Web server or some type of Web application framework. At a high level, the CMS has two primary roles: (1) request the client's credentials (called a *network ID*) and (2) respond with the host access credentials, which consist of the *host ID* and a password or passticket, depending on the type of HCM. In order to carry out the request and response process, the CMS calls upon the Network Security plug-in to acquire the user's network ID from the network security application. Then, the

CMS calls upon the HCM to acquire the user's host access credentials. It then returns the host access credentials to the Host On-Demand client in the form of an XML document.

The login macro is recorded 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). In broad terms, it 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.

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 Digital Certificate Access Server (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 requires you to write your own plug-in. For more information, refer to Chapter 7, "Customizing Web Express Logon," on page 53.

Figure 1 illustrates the overall flow of macro-based automation by showing you the key components discussed above and how they interact together to achieve logon automation:

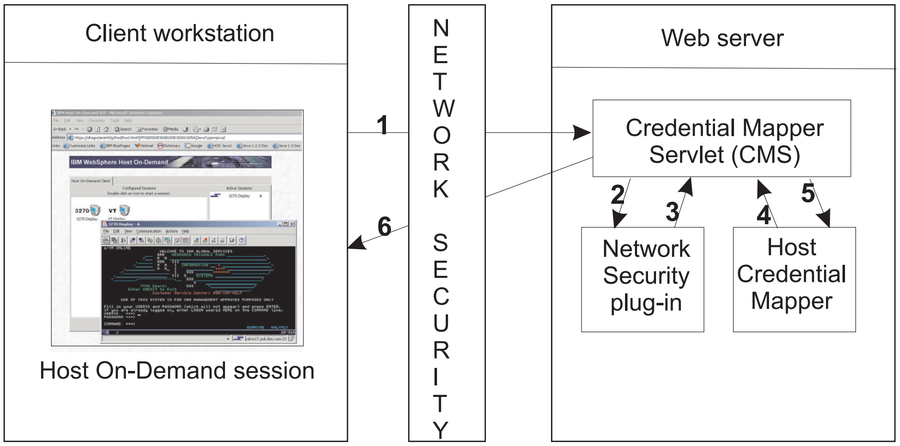


Figure 1. Macro-based automation

The following are the steps that take place 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. The numbers in the list correspond to the numbers in Figure 1:

1. The client, who has been authenticated by the network security application, sends an HTTPS request to the CMS to obtain the host credentials.
2. The CMS requests the user's network ID from the Network Security plug-in.
3. The Network Security plug-in responds to the CMS with the user's network ID.
4. The CMS passes the network ID to the HCM.
5. The HCM returns the host ID along with a password or passticket, depending on the HCM type.
6. The CMS returns the host credentials 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.

Macro-based automation has been successfully tested with the following applications:

- IBM Tivoli Access Manager for e-business Version 4.1
- Microsoft Active Directory
- Netegrity Siteminder Version 5.5
- WebSphere Application Server Versions 4 and 5
- IBM DB2 Universal Database Version 7
- z/OS V1R4 with APAR PQ74454

The macro-based automation version of Web Express Logon can function with other applications that are not listed here.

Connection-based automation (iSeries only)

Connection-based automation works in iSeries environments that meet the following criteria:

- operate within a Windows Domain
- have Kerberos-based network authentication enabled
- run OS/400 version 5R2 or later (iSeries)
- run one or more of the following client operating systems:
  - Windows 2000 (Professional, Server, and Advanced Server)
  - Windows XP Professional
  - Windows Server 2003

Working in conjunction with Kerberos-based network authentication and an IBM technology called Enterprise Identity Mapping (EIM), this iSeries environment already has the capability to provide single sign-on. Web Express Logon simply extends this capability by allowing Host On-Demand to use the existing methodology for acquiring credentials to allow users to bypass the host session login screen. Because it works within your existing environment, connection-based automation does not require the use of a CMS, a login macro, the Network Security plug-in, nor the HCM.

With connection-based automation, Host On-Demand sessions allow users to bypass the logon screen by using the user ID and passticket credentials obtained when users connect to target iSeries systems from client workstations that are on a Windows Domain. The Enterprise Identity Mapping (EIM) feature of OS/400 calculates the target iSeries user profile to use, and the network authentication service of OS/400 defines the Kerberos realms to trust.

Connection-based automation relies on the following three key components:

- Windows domain controller
- Key Distribution Center (KDC)
- Kerberos passticket

The Windows domain controller gives users access to the network, and the Key Distribution Center (KDC) gives users access to individual resources within that network. The KDC gives users access to these resources by granting them Kerberos passtickets. To illustrate this concept with an analogy, think of an individual attempting to enter a building. Once the user is authenticated to enter the building (by the domain controller), he attempts to enter a room within the building. However, at this point, he is challenged to provide additional credentials. He requests access to the room (from the KDC) and is then authenticated to enter the room (with a Kerberos passticket).

Figure 2 illustrates the overall process of connection-based automation in an iSeries environment with Kerberos authentication enabled:

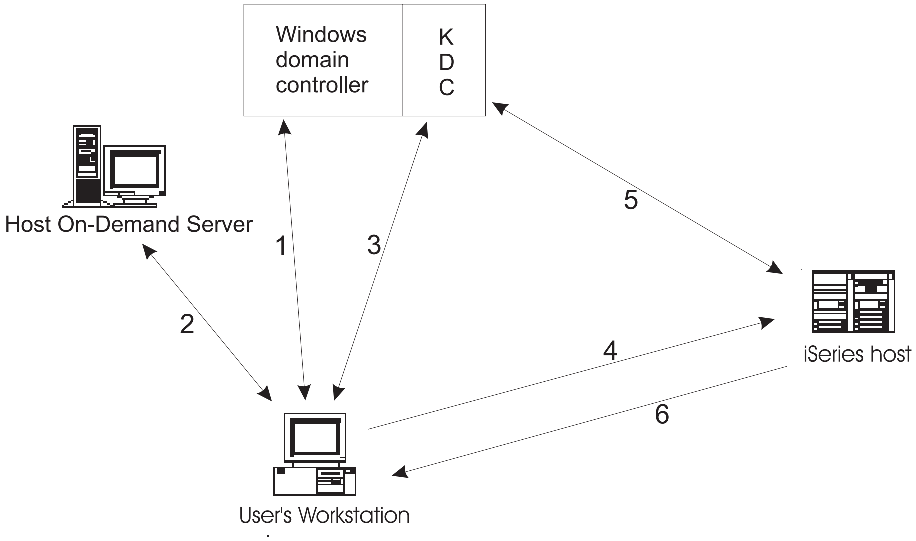


Figure 2. Connection-based automation

The following list shows you the steps that take place during connection-based automation. The numbers in the list correspond to the numbers in Figure 2:

1. A user logs on to the Windows domain controller.
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 passticket from the Key Distribution Center (KDC).
4. The user attempts to create a connection with the identified session using the Kerberos Passticket as the credential.
5. The iSeries host validates the Passticket with the KDC.
6. The user is successfully logged in

Part 3. Planning

Chapter 3. Planning for implementation

Having a clear understanding of your environment and how you plan to implement Web Express Logon in your environment will save you valuable time in the implementation phase. Be sure that you take time to develop your strategy and gather the necessary resources and skills. A firm plan is key to a successful implementation.

We recommend that you begin planning by taking the following steps:

Step 1: Choose your style of logon automation.

As described in the introduction, Host On-Demand offers two styles of logon automation: macro-based automation and connection-based automation. The style of logon automation that best suits your environment depends on your host type. Macro-based automation is for environments of varying host types that *are not* using Kerberos authentication. Connection-based automation, on the other hand, is *only* for IBM eServer iSeries host environments that support Kerberos authentication.

Step 2: Identify areas of credential challenges.

Credential challenges are the times at which users are prompted to provide IDs and passwords. The first step is to evaluate your existing network infrastructure and identify which credential challenges exist for your users. Approach this step by simulating a typical day and identifying all the points at which users are prompted to provide credentials. For example, in a corporate environment, users may have to provide credentials when attempting to access any of the following resources:

- operating system
- corporate home page
- Web-based applications
- host-based applications

Step 3: Take an inventory of your environment.

At this point, you should know which style of logon automation is appropriate for your environment and what components are necessary to implement Web Express Logon. Before you can successfully plan your deployment strategy and estimate the scope of implementation, take a moment to take an inventory of your environment and answer the following questions according to your style of logon automation:

Macro-based automation

- What is your host type?
- What is your network security application? IBM Tivoli Access Manager? Netegrity Siteminder? Microsoft Active Directory? Other?
- If your network security application is not one of the three applications listed in the previous question, you will need to customize your own Network Security

plug-in. Do you have someone on hand who has some J2EE knowledge and experience working with J2EE-compliant servlets?

- What Web application server are you using? IBM WebSphere Application Server? BEA WebLogic? Apache Tomcat?
- Do you have a J2EE-compliant Web application server to deploy the Credential Mapper Servlet (CMS) to your Web server?
- What will you use as your Host Credential Mapper (HCM)? IBM DB2? LDAP?
- Do you plan to use DCAS on a z/OS platform, or do you plan to use a vault-style database to acquire the host access credentials?

Connection-based automation

- What level of OS/400 are you running on your iSeries host or hosts? It must be version 5R2 or later in order to use Web Express Logon.
- Are your Host On-Demand clients authenticated using Windows Domain?
- What are you using as your Key Distribution Center (KDC)?
- Are your clients running one or more of the following operating systems?
  - Windows 2000 Professional, Server, or Advanced Server
  - Windows XP Professional
  - Windows 2003 Server

  If not, you will need to upgrade, since other versions of Windows do not support Kerberos authentication.

Step 4: Develop your deployment strategy.

Now that you have evaluated your need for a Web Express Logon solution, chosen the style of logon automation that best works in your environment, and taken an inventory of your company's environment and resources, you can begin developing your deployment strategy. Consider issues such as how many/which users will be affected by this implementation, which skills are required for a successful implementation, and how many people you will need to participate in the setup process.

Step 5: Establish an HCM (macro-based automation only).

This step does not apply to iSeries platforms that support Kerberos authentication.

If you are in an environment that *does not* support Kerberos authentication, you must have an HCM in place. An HCM is a back-end repository that associates 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 requires you to write your own plug-in. For more information, refer to Chapter 7, "Customizing Web Express Logon," on page 53. The CMS queries this repository during the logon process.

Part 4. Implementation

Chapter 4. Macro-based automation

Host On-Demand provides a Credential Mapper Servlet (CMS) that supports three network security applications:

- IBM Tivoli Access Manager for e-business Version 4.1
- Netegrity Siteminder 5.5
- Microsoft Active Directory (Windows Domain)

If you do not have one of these three network security applications, you will need to customize your own version of the CMS. For more information, refer to Chapter 7, "Customizing Web Express Logon," on page 53.

Step 1: Configure the Credential Mapper Servlet (CMS).

Take the following steps when using one of the three versions of the CMS shipped with Host On-Demand. They are packaged as individual WAR files on the Host On-Demand Version 8 CD.

A. Locate the WAR files on the Host On-Demand Version 8 CD

The three WAR files are located in the `cdimage\apps\wel` subdirectory. Choose the one that matches your network security application:

- IBM Tivoli Access Manager: amcms.war
- Netegrity Siteminder: smcms.war
- Microsoft Active Directory (Windows Domain): wmcms.war

B. Unpack the WAR file.

Once you select the WAR file that matches your environment, unpack it and view its contents. In addition to several CLASS files, you will see 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 future steps. 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 Chapter 8, "Troubleshooting Web Express Logon," on page 63.

C. Become familiar with the INIT parameters in the web.xml file.

The web.xml file contains three default INIT parameters. INIT parameters are what the CMS configures to work in your environment. They adapt the servlet to your environment.

- **Network Security plug-in**: This value is the full path name of the class that handles the CMS interface into the network security application. This example is taken from the amcms.war file, which is based on the Tivoli Access Manager network security application:

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

The Network Security plug-in does not apply to the Microsoft Active Directory (Windows Domain) web.xml file. This is because the Windows login ID is used as the network ID.

- **Host Credentials plug-in**: This is a compound value that contains the list of all available credential mappers, for example, CMPIDCASPlugin and CMPIVaultPlugin. Before you edit the code, however, you will notice that the value is echo.

Using this echo parameter is optional and can help you confirm that you've deployed your servlet correctly in later steps.

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

- **echo plug-in**: The credential mapper value (echo) for the previous plug-in is the name of the parameter for this plug-in. They must match. Once you are ready to edit your code, you will need to replace the echo value for the Host Credentials plug-in as well as the name of this parameter with the name of the credential mapper.

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

D. Edit the web.xml file

Now that you have viewed the CMS-related INIT parameters, you are ready to edit the web.xml file.

In the web.xml file, *do not* change the following:

- the name of the servlet (recommended)
- the CMPINetworkSecurity parameter name or value (amcms.war and smcms.war files only)
- the CMPICredentialMappers parameter name (you will need to change this parameter value, however)

Edit the CMS-related parameters

To edit the CMS-related parameters, take the following steps:

1.

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\HODWEL.log on a Windows platform.

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.

2. Locate the CMPICredentialMappers parameter and change the name of its value to something that appropriately represents the function of the plug-in. This is a compound value that contains the list of all available HCMs. Each value is separated with a comma character. In the following example, we have added two parameter values, one for DCAS and the other for Vault. You will most likely have only one parameter value, however.

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

3. Add a new parameter for each of the new parameter values that you specified in the previous step.

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

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

The parameter value is a compound value that contains the full class path name of the implementing class, the authentication type to be addressed by the credential mapper, and the host mask. The values are separated by the comma character.

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 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 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 credential mapper to select to handle the request. You can pair multiple authentication types together to give HCMs the freedom to support multiple authentication types. Use the vertical bar character to join multiple authentication types.
The five identified authentication types are listed in Table 2:

Table 2. Authentication types and descriptions

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

Host mask
The host mask is a secondary selection criteria used by the CMS to identify the most appropriate credential mapper. 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.
Table 3 lists valid wild-carded addresses:

Table 3. Host masks and values matched

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`

Add DCAS parameters for CMPIDCASPlugin.

For solutions that use z/OS and DCAS, add the DCAS plug-in parameters. Adding these parameters allows the HCM to map the user's network ID to his host ID and then get a passticket from the DCAS application running on the host. A passticket is a credential that is similar to a password, however a passticket expires after a certain amount of time and is used only one time. DCAS requires a Security Access Facility (SAF)-compliant server product, such as an IBM Resource Access Control Facility (RACF) security server, that supports passticket generation.

To use the DCAS plug-in, you must configure the DCAS server. To configure the DCAS server, 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 PQ74454 for information on how to configure the DCAS server to function with Web Express Logon.

Use the DCAS.xml file located in the WAR file as a reference for adding parameters when editing the web.xml file.

Required DCAS parameters: The following two Host Credential plug-in parameters allow the client to connect to the DCAS server securely:

CMPI_DCAS_KEYRING_FILE

This parameter specifies a keyring database. A keyring must be specified to provide 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. This parameter is a file reference to the keyring to be used. The DCAS plug-in is the DCAS client.

You will create this file in "Step 3: Create SSL key database (DCAS only)." on page 26.

CMPI_DCAS_KEYRING_PASSWORD

This parameter specifies the password for the keyring database.

This parameter should be encrypted using the password encryption tool. It is decrypted by the HCM before using it. For more information about the password encryption tool, refer to Appendix A, "Password encryption tool," on page 71.

The following parameters are designed to work your JDBC database credential mapper. Using this type of network-accessible database provides you with a flexible and secure means of associating users' network IDs to their host IDs. By storing all the relevant access information in this web.xml file, you can configure access to an existing database or point to a newly created database. The level of security for the database varies according to database vendor.

If you are using LDAP as your credential mapper, you will need to create your own HCM using Chapter 7, "Customizing Web Express Logon," on page 53.

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:6789/CMTEST.

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.

CMPI_DCAS_DB_USERID

This is the ID of the user account to use when accessing the database.

CMPI_DCAS_DB_PASSWORD

This is the password of the user account to use when accessing the database.

This parameter should be encrypted using the encrypt password tool. It is decrypted by the HCM before using it. For more information about the password encryption tool, refer to Appendix A, "Password encryption tool," on page 71.

CMPI_DCAS_DB_TABLE
This entry identifies the table to use for the needed query.

The following four parameter values should match the column names in your credential mapper database and should clearly indicate the contents of the columns. With some databases, such as IBM DB2, the four column headings in the database must be in all upper case, for example, NETWORKID, HOSTADDRESS, APPLICATIONID, and HOSTID.

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

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

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

CMPI_DCAS_DB_HOSTID_COL_NAME
This entry identifies the name of the column that contains the user’s host identification value (HOSTID).

Based on the information provided by the parameters above, you can make an SQL query of the database to get the host ID. This query uses the network ID, the host address, and the host application as keys for the query. The result is identified in the Host Identification column. Assuming that the query is successful, a call is made to the DCAS server to request the passticket.

Optional DCAS parameters: The following DCAS parameters are optional:

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.

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.

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.

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.

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.

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.

Add Vault parameters for CMPIVaultPlugin.

For environments that use JDBC-based Vault host security, add the Vault plug-in parameters. This model is identical to the database mechanism used to associate network IDs and host IDs in the DCAS passticket environment. The only difference is that Vault-style authentication requires the CMPI_VAULT_DB_HOSTPW parameter in the web.xml file.

Use the Vault.xml file located in the WAR file as a reference for adding parameters when editing the web.xml file.

Required Vault parameters: The following Vault parameters are required:

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:6789/CMTEST.

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.

CMPI_VAULT_DB_USERID

This is the ID of the user account to use when accessing the database.

CMPI_VAULT_DB_PASSWORD

This is the password of the user account to use when accessing the database.

This parameter should be encrypted by the encrypt password tool. It is decrypted by the HCM before using it. For more information about the password encryption tool, refer to Appendix A, “Password encryption tool,” on page 71.

CMPI_VAULT_DB_TABLE

This entry identifies the table to use for the needed query.

The following five parameter values should be in all upper case and should exactly match the column names in your credential mapper database. With some databases, such as IBM DB2, the five column headings in the database must be in all upper case, for example, NETWORKID, HOSTADDRESS, APPLICATIONID, HOSTID, and PASSWORD.

CMPI_VAULT_DB_NETID_COL_NAME

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

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

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

CMPI_VAULT_DB_HOSTID_COL_NAME
This entry identifies the name of the column that contains the user's host identification value (HOSTID).

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

Based on the information provided by the parameters above, you can make an SQL query of the database to get the host ID. This query uses the network ID, the host address, and the host application as keys for the query. The result is identified in the Host Identification column. Assuming that the query is successful, the user ID and password are returned.

Optional Vault parameters: The following Vault parameters are optional:

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.

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.

Once you have completed editing the web.xml file, you may need to repackage the WAR file. This process depends on the requirements of your Web application server. Consult your product's documentation to learn more about these requirements.

Step 2: Deploy the CMS.

At this point, you are ready to deploy the servlet to the Web server. Refer to your Web server application's documentation for details of how to deploy the servlet.

Step 3: Create SSL key database (DCAS only).

In order to communicate with a DCAS server, an SSL connection must be established using client authentication. This requires you to create a key database file, for example HODDCAS.p12. To create the file, use the Host On-Demand Certificate Management GUI on Windows and AIX platforms, or use a P12 keyring tool for other platforms. This key database file must contain the DCAS client's personal certificate and the DCAS server's certificate (public key) information. Also, the DCAS client certificate must be added/imported to the DCAS server's keyring for SSL client authentication.

For more information about creating this key database file, refer to the Planning, Installing, and Configuring Host On-Demand guide, which is located in the Host On-Demand InfoCenter at Start > Programs > IBM WebSphere Host On-Demand > InfoCenter or on the Web at http://www.ibm.com/software/webservers/hostondemand/library/v8infocenter/.

To create a keyring database called HODDCAS.p12 file that will be specified in the CMPI_DCAS_KEYRING_FILE parameter in your web.xml file, take the following steps on a Windows machine

1. Click Start > Programs > IBM WebSphere Host On-Demand > Administration > Certificate Management.
2. Click Key Database File and select New. For the Key database type, select PKCS12. For File Name, type HODDCAS.p12). For Location, type C:\Program Files\IBM\HostOnDemand. (Note that you may choose a different name and location.)
3. Click OK.
4. Type the password and make a note of it.
5. Click OK.
6. Add the DCAS server's certificate to the key database. Be sure that the key database content is for the signer certificate. If it is not, select the pull-down menu and change it. Then select Add.
7. Select Binary DER data for the data type. If the server certificate is in ASCII format, select Base64-encoded ASCII data.
8. Type the file name in the For Certificate File Name field.
9. Type the path name in the Location field.
10. Click OK.
11. Enter a label for the certificate and click OK.
12. Add the DCAS client's certificate to the key database.
13. Change the Key database content to Personal Certificates and click Export/Import.
14. Select Import Key as the Action Type.
15. Select PKCS12 for the Key file type.
16. Type the client certificate's p12 file name in the File Name field and the path name in the Location field.
17. Click OK and enter the client certificate PIN.
18. Click OK.
19. Exit the Certificate Management GUI.

Step 4: Use the Deployment Wizard to create your HTML file.

The Host On-Demand Deployment Wizard allows you to specify how sessions are defined and managed. You can choose from three different configuration models:

- HTML-based model
- Configuration server-based model
- Combined model

If you are using the HTML-based or Combined models, you can create your HTML file as normal. However, for the Configuration server-based model, you must configure the HTML file with additional steps:

1. When you reach the Logon Type window in the Deployment Wizard, you must select one of three options:
   - **Prompt users to enter Host On-Demand user ID**: Select this option only if you want users to be challenged for their credentials.
   - **Use Web Express Logon**: Select this option to use the network ID to map to a 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
     ```

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

     a. In the web.xml file, update the following INIT parameter with the new Vault credential mapper name, for example, `CMPIConfigServer_`:

        ```
        <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`:

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

     b. In the web.xml file, configure the remaining Vault parameters *except* these two parameters:
        - CMPI_VAULT_DB_HOSTADDR_COL_NAME
        - CMPI_VAULT_DB_HOSTAPP_COL_NAME

        Use the section “Add Vault parameters for CMPIVaultPlugin.” on page 25 as a reference. You will need to prepend the new credential mapper name to the Vault parameter names, for example, CMPIConfigServer_CMPI_VAULT_DB_ADDRESS.

     c. In your Vault credential mapper database, create a new table with three columns, for example:
        - NETWORKID
        - HODID
        - PASSWORD

        Be sure that the NETWORKID 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.

- **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 (wmcms.war).

2. 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
   ```

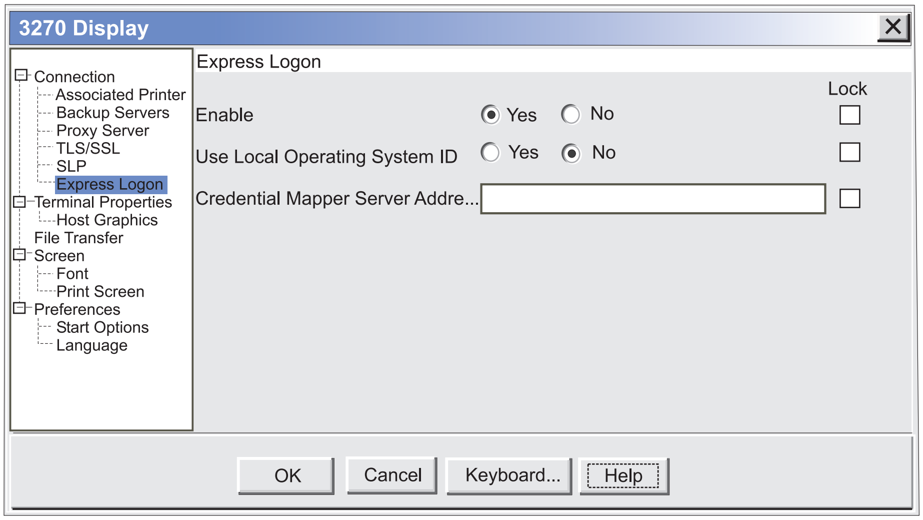
   where *myhodserver* is the machine you are pointing to with the *junction_name*.

Step 5: Configure the Host On-Demand session.

You must configure your session properties to use Express Logon. You can do this in the following two ways:

- Right-click a session icon and select Properties. On the left side of the window, select Express Logon under Connection.
- In the Deployment Wizard on the Host Sessions window, highlight your session and select Properties under the Configure drop-down menu. On the left side of the window, select Express Logon under Connection.

The options on the Express Logon window differ depending on the type of session you are configuring. Figure 3 is the Express Logon window for 3270 sessions:



Express Logon

Lock

Enable ● Yes ○ No ☐

Use Local Operating System ID ○ Yes ● No ☐

Credential Mapper Server Addre... ☐

Figure 3. 3270 Express Logon

Once you select Yes to enable Express Logon, 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 (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, the code in your XML should look like the following:

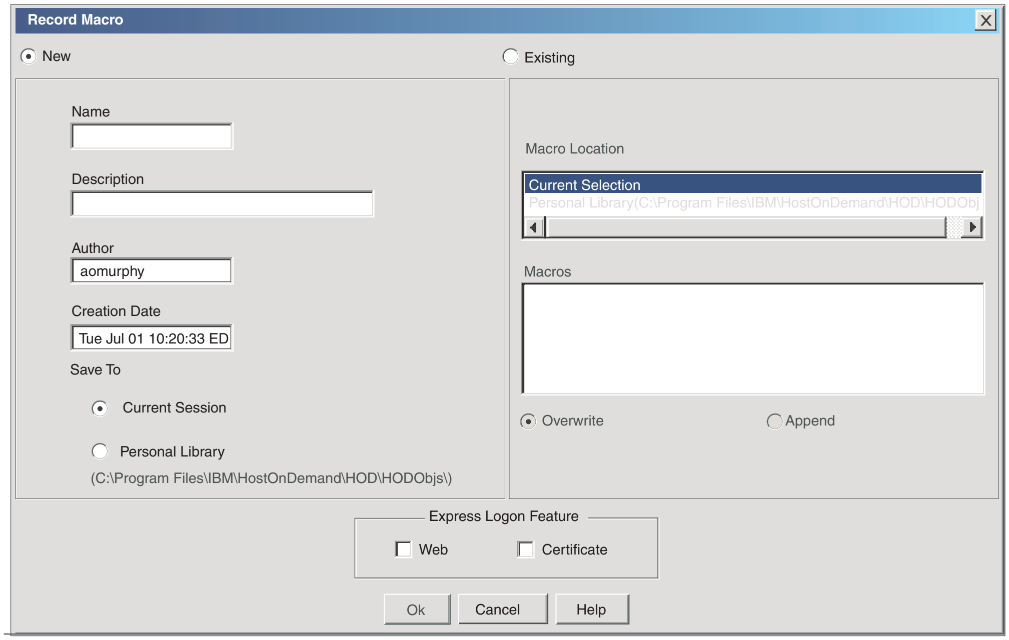
```
<servlet>
      <servlet-name>CredMapper</servlet-name>
      <display-name>CredMapper</display-name>
      <servlet-class>com.ibm.eNetwork.security.sso.cms.CredMapper</servlet-class>
```

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.

Step 6: Record the macro.

Take the following steps to record the automation macro:

1. Open your session and click the Record macro button on the toolbar. On the Record macro window (Figure 4), select Web under Express Logon Feature.



● New ○ Existing

Author: aomurphy

Creation Date: Tue Jul 01 10:20:33 ED

Save To
● Current Session
○ Personal Library
(C:\Program Files\IBM\HostOnDemand\HOD\HODObjs\)

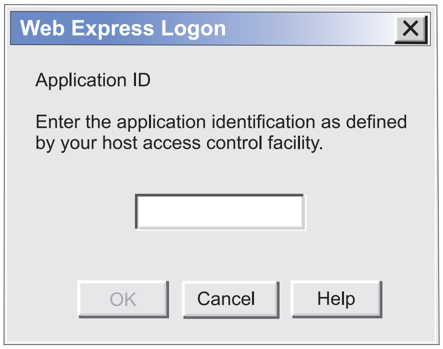
Macro Location: Current Selection
Personal Library(C:\Program Files\IBM\HostOnDemand\HOD\HODObj

● Overwrite ○ Append

☐ Web ☐ Certificate

Figure 4. Record macro window

2. Enter the application ID (3270 sessions only) in the Application ID window (Figure 5) 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.

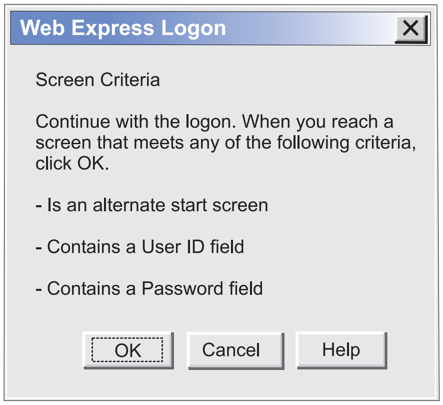


Application ID

Enter the application identification as defined by your host access control facility.

Figure 5. Application ID

3. The Screen Criteria window (Figure 6) 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..



Screen Criteria

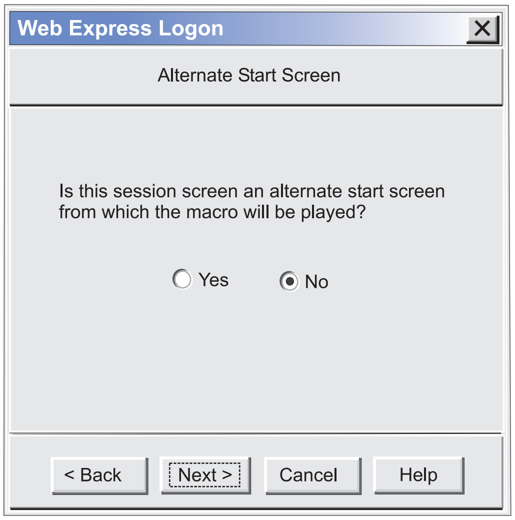
Continue with the logon. When you reach a screen that meets any of the following criteria, click OK.

- Is an alternate start screen
- Contains a User ID field
- Contains a Password field

Figure 6. Screen Criteria

4. On the Alternate Start Screen window (Figure 7), specify whether this screen is an alternate start screen and then 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.

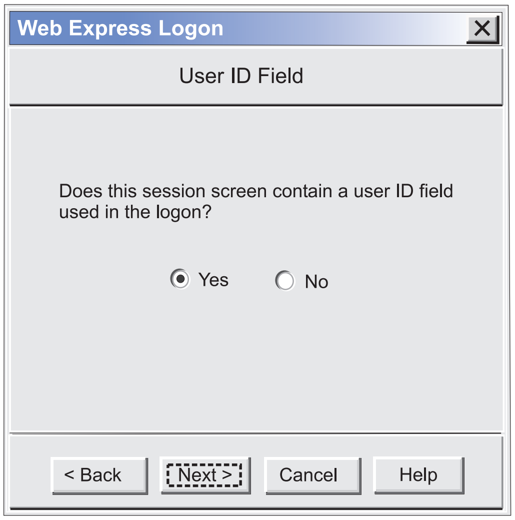


Is this session screen an alternate start screen from which the macro will be played?

○ Yes ● No

Figure 7. Alternate Start Screen

5. On the User ID Field window (Figure 8), select Yes to specify that the session screen contains a user ID field. Click Next.



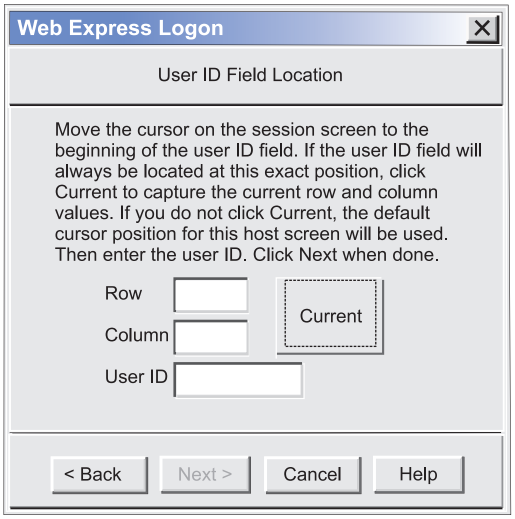
Does this session screen contain a user ID field used in the logon?

● Yes ○ No

Figure 8. User ID Field

6. On the User ID Field Location window (Figure 9), 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.

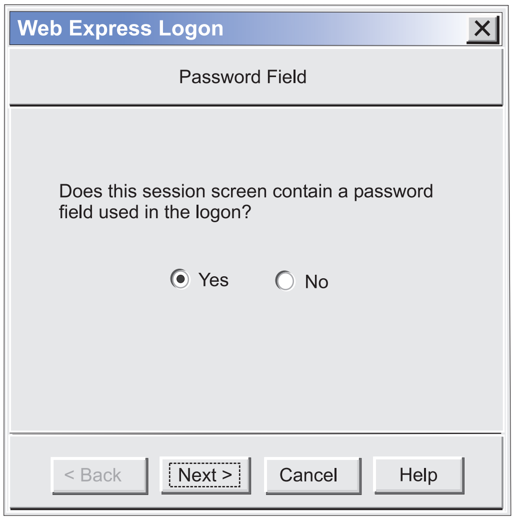


Move the cursor on the session screen to the beginning of the user ID field. If the user ID field will always be located at this exact position, click Current to capture the current row and column values. If you do not click Current, the default cursor position for this host screen will be used. Then enter the user ID. Click Next when done.

Row
Column
User ID

Figure 9. User ID Field Location

7. On the Password Field window (Figure 10), select Yes to specify that the session screen contains a password field. Click Next.



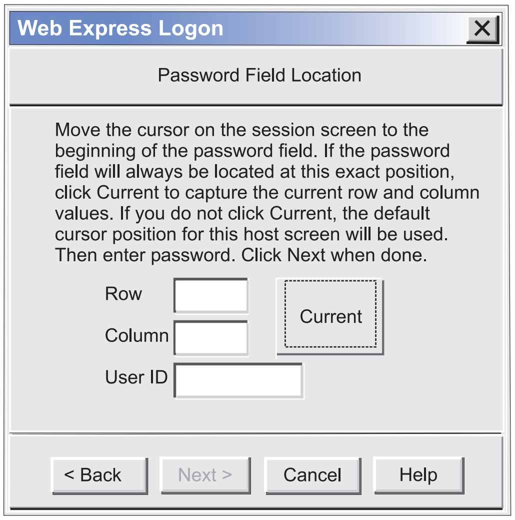
Does this session screen contain a password field used in the logon?

● Yes ○ No

Figure 10. Password Field

8. On the Password Field Location window (Figure 11), 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.

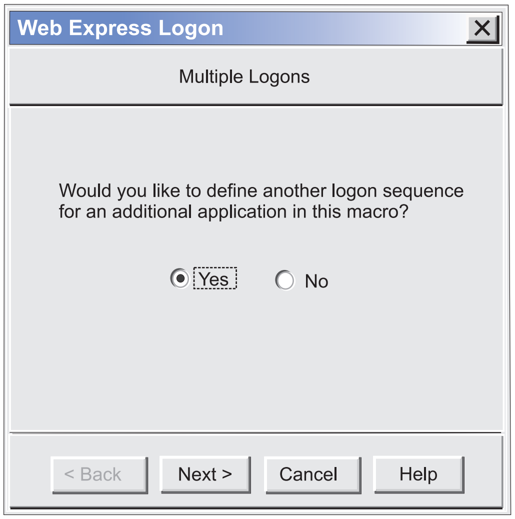
Click Current only if you will not be using this screen for multiple applications and the location of the password field never changes.



Move the cursor on the session screen to the beginning of the password field. If the password field will always be located at this exact position, click Current to capture the current row and column values. If you do not click Current, the default cursor position for this host screen will be used. Then enter password. Click Next when done.

Figure 11. Password Field Location

9. On the Multiple Logons window (Figure 12), click either Yes or No.

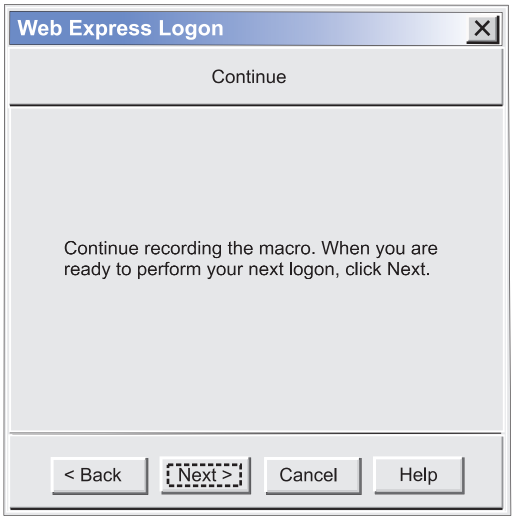


Would you like to define another logon sequence for an additional application in this macro?

● Yes ○ No

Figure 12. Multiple Logons

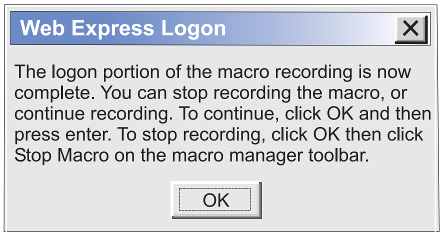
Click Yes if you want to define another logon sequence for an additional application. If you click Yes, you will advance to the Continue window (Figure 13). Once you reach a point in the macro that requires another User ID and password logon, click Next.



Continue recording the macro. When you are ready to perform your next logon, click Next.

Figure 13. Continue

Click No to finish recording the logon portion of the macro (Figure 14). Click OK to finish.



The logon portion of the macro recording is now complete. You can stop recording the macro, or continue recording. To continue, click OK and then press enter. To stop recording, click OK then click Stop Macro on the macro manager toolbar.

Figure 14. Finish

10. Finish recording your macro using the Macro Manager, and click Stop on the toolbar.
11. 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.

Chapter 5. Connection-based automation

Unlike macro-based automation, connection-based automation does not require the use of a Credential Mapper Servlet (CMS), a login macro, the Network Security plug-in, nor the Host Credential Mapper (HCM). Instead, it extends the existing single sign-on capability of iSeries environments that meet the following criteria:

- operate within a Windows Domain
- have Kerberos-based network authentication enabled on each target iSeries system
- run OS/400 version 5R2 or later (these versions support Kerberos-based network authentication)
- run one or more of the following client operating systems:
  - Windows 2000 (Professional, Server, and Advanced Server)
  - Windows XP Professional
  - Windows Server 2003

You must configure your iSeries environment to use single sign-on capability in order to implement connection-based logon automation.

The iSeries environment provides single sign-on capability by working in conjunction with Kerberos-based network authentication 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 host session login screen.

Both EIM technology and Kerberos are available with Version 5R2 of the OS/400 operating system. EIM is an IBM infrastructure technology that allows you to manage multiple user identities and user registries easily and inexpensively while maintaining secure authentication and authorization. This architecture describes the relationships between individuals or entities in an enterprise and the many identities that represent them within the enterprise. Kerberos, on the other hand, is a network authentication protocol that identifies and authenticates users who request to log on to a network. Together, EIM and Kerberos provide single sign-on capability.

Although this document does not instruct you how to configure your iSeries environment for single sign-on capability, the following resources are available to help you:

- IBM eServer iSeries Enterprise Identity Mapping document: http://publib.boulder.ibm.com/iseries/v5r2/ic2924/info/rzalv/rzalv.pdf
- Enterprise Identity Mapping Web site: http://www.ibm.com/servers/eserver/security/eim/
- IBM eServer iSeries Resource Library: http://www.ibm.com/eserver/iseries

Once you have configured your iSeries environment to use single sign-on capability, you are ready to configure Host On-Demand to extend this single sign-on capability. To accomplish this, take the following two steps:

Step 1: Use the Deployment Wizard to create your HTML file.

The Host On-Demand Deployment Wizard allows you to specify how sessions are defined and managed. You can choose from three different configuration models:

- HTML-based model
- Configuration server-based model
- Combined model

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:

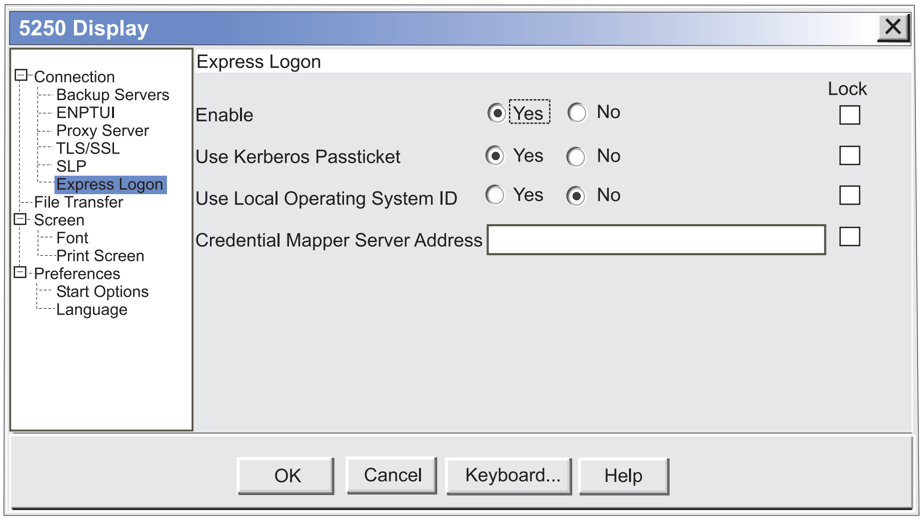
```
ConfigServer=myhodserver.ibm.com
```

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

Step 2: Configure your Host On-Demand session.

Configure your 5250 session properties to use a Kerberos passticket. You can do this in the following two ways:

- Right-click a session icon and select Properties. On the left side of the window, select Express Logon under Connection. On the Display window, select Yes to enable Express Logon and Yes to Use Kerberos Passticket (Figure 15).
- In the Deployment Wizard on the Host Sessions window, highlight your session and select Properties under the Configure drop-down menu. On the left side of the window, select Express Logon under Connection. On the Display window, select Yes to enable Express Logon and Yes to Use Kerberos Passticket (Figure 15).



Express Logon

Lock

Enable ● Yes ○ No ☐

Use Kerberos Passticket ● Yes ○ No ☐

Use Local Operating System ID ○ Yes ● No ☐

Credential Mapper Server Address ☐

Figure 15. 5250 Express Logon

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.

Part 5. Real-life scenario

Chapter 6. Scenario: Macro-based automation in a z/OS/3270 environment

Overview

AC Gas, Inc. is a fictitious electric and gas company based in a metropolis area. It has over 1000 employees who must connect to host systems and host-based applications throughout the day to access customer and employee records. Using IBM WebSphere Host On-Demand, these employees 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.

Business problem

The company's employees are constantly logging in and out of host sessions and Web applications, forcing them to keep up with several IDs and passwords. They complain that they are continuously prompted to provide their credentials, something they feel takes up valuable time. Also, the company's IT staff is complaining because they receive several calls a day from employees who have forgotten or misplaced their passwords.

Every year, the company executives send out a satisfaction survey to its employees. On this survey, one of the top complaints is the need to log in multiple times per day and having to keep up with multiple IDs and passwords, many of which are different. The executives also noticed that another big complaint is the amount of time employees spend calling the support line to have representatives reset lost passwords. The executives investigate even further and find out that password-related support calls make up over 40% of the total amount of calls, something that is costing the company a sizable amount of money. They immediately begin researching ways to eliminate this costly multiple logon problem. They find several applications that claim to offer a single sign-on solution, but none of them function within the company's host-based infrastructure. They decide to investigate Host On-Demand's Web Express Logon. They start by planning their strategy.

Planning

1. **Choose the style of logon automation**: AC Gas, Inc. has a z/OS/3270 emulation environment, and therefore uses macro-based automation.
2. **Identify areas of credential challenges.**: They evaluate their existing network infrastructure and decide which areas would most benefit from logon automation. They identify which logon processes slow productivity and cause the most frustration for users, and then they make a list of which host types and emulation types the logon processes use. They know that having this list handy will later help them estimate the scope of implementing Web Express Logon.
3. **Take inventory of environment**: Their next step in the planning phase is to take inventory of the company's environment. Realizing that the success of Web Express Logon depends on understanding how these components fit together, including how the network security application performs authentication and authorization, they set out to learn how these pieces will play a role in the automated logon process.

Table 4. Inventory of environment

Component	AC Gas, Inc.'s environment
Web Application Server	IBM WebSphere Application Server Version 5.0
Database	IBM DB2 Universal Database Version 7
Network security application	IBM Tivoli Access Manager for e-business Version 4.1
Host OS	z/OS V1R4 with APAR PQ74454
Web server	IBM HTTP Server
Host authentication	DCAS

4. **Develop deployment strategy**: AC Gas, Inc. now begins thinking about how many users will be affected by implementing Web Express Logon and where they plan to install the servlet. They know that only those users who authenticate through their network security application (Tivoli Access Manager) can take advantage of this feature. They begin documenting which users are eligible and making plans to install the feature on the Web server.
5. **Establish a Host Credential Mapper (HCM)**: Since the DCAS and Vault parameters supplied with Web Express Logon are designed to work with a JDBC database, AC Gas, Inc. use IBM DB2 as their HCM. Recall that the JDBC database is the repository that maps the user's network ID to his host ID in order to achieve logon automation. In the database, they create a table with the following column headings: NETWORKID, HOSTADDRESS, APPLICATIONID, and HOSTID. Since IBM DB2 is case sensitive, all the column headings in the database are in upper case. Later, when they edit the web.xml file, they will make sure that the parameter values for the column headings match.

Consult the documentation for SQL calls for your database platform.

Implementation

Once AC gas, Inc. plans their strategy, they move directly into the implementation phase:

Step 1: Configure the Credential Mapper Servlet (CMS)

AC Gas, Inc. understands that the Credential Mapper Servlet (CMS) is the core of the credential-mapping framework and must reside on a Web server. Using the Host On-Demand Version 8 CD, they browse to the `cdimage\apps\wel` subdirectory and locate the amcms.war file. This is the WAR file that is designed to work with Tivoli Access Manager. Next, they unpack the amcms.war file and view its contents in an effort to become more familiar with how it is organized. Their next step is to edit the web.xml file.

Before editing the web.xml file, they gather the list of parameters from Chapter 4, "Macro-based automation," on page 19 and become familiar with which ones are required and which ones are optional. The following is their web.xml file after customizing it:

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

    <init-param>
      <param-name>CMPICredentialMappers</param-name>
      <param-value>CMPIDCASPlugin </param-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>CMPIDCASPlugin</param-name>
      <param-value>com.ibm.eNetwork.security.sso.cms.CMPIDCAS,
               AuthType_3270Host,*</param-value>
    </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_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_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>&*$^%&***</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_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>
```

Now that the administrator has edited the web.xml file, he repackages the WAR file using a zip tool.

Step 2: Deploy the CMS.

AC Gas, Inc. uses WebSphere Application Server to deploy the CMS, which is made up of server-side Java classes that are deployed, managed, and executed on a J2EE-compliant application server. On a Windows platform, they take the following steps:

1. Click Start > Programs > IBM WebSphere > Application Server v5.0 > Administrative Console to open the Administrative Console.
2. In the left navigation field, click Applications > Install New Application
3. Select Local path and browse to the amcms.war file.

4. Once the local path field is filled, specify the Context Root. 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`.
5. Click Next.
6. Check the Generate Default Bindings box. By choosing this option, you can jump directly to the Summary step and deploy the WAR file.
7. Click Next to go to the Step 1. Provide options to perform the installation..
8. Scroll down and click the Step 4 Summary link.
9. Click Finish.
10. On the left navigation field, click Applications > Enterprise Applications.
11. Scroll through the applications and check the checkbox beside amcms_war, which is your application name.
12. Click Start at the top to start the application.
13. Test to make sure that the WAR file installed correctly by pointing a browser to http://server_name:9080/wel/CredMapper. Port 9080 is the default WebSphere port. You can also go through Tivoli Access Manager (port 80) at this URL: http://server_name/WAS/wel/CredMapper.
14. Once you type the URL in a browser, make an HTTPS request to the servlet and get a response via an XML file. Host On-Demand returns the XML code. If you see the code, the request was successfully returned.

Step 3: Create SSL Key database

Since AC Gas, Inc. has a DCAS-—based host, they need to create an SSL Key database using the Host On-Demand Certificate Management.

To create an SSL Key database on Windows and AIX platforms, use the Host On-Demand Certificate Management. For other platforms, use a P12 keyring tool. For more information, refer to the Planning, Installing, and Configuring Host On-Demand guide, which is located in the Host On-Demand InfoCenter at Start > Programs > IBM WebSphere Host On-Demand > InfoCenter or on the Web at http://www.ibm.com/software/webservers/hostondemand/library/v8infocenter/.

In order to communicate with a DCAS server, an SSL connection must be established using client authentication. This key database file must contain the DCAS client’s personal certificate and the DCAS server’s certificate (public key) information. Also, the DCAS client certificate must be added/imported to the DCAS server’s keyring for SSL client authentication.

Company ABC creates a key database called HODDCAS.p12 file on a Windows machine by taking the following steps:

1. Click Start > Programs > IBM WebSphere Host On-Demand > Administration > Certificate Management.
2. Click Key Database File and select New. For the Key database type, select PKCS12. For File Name, type HODDCAS.p12). For Location, type C:\Program Files\IBM\HostOnDemand. (Note that you may choose a different name and location.)
3. Click OK.
4. Type the password and make a note of it.
5. Click OK.

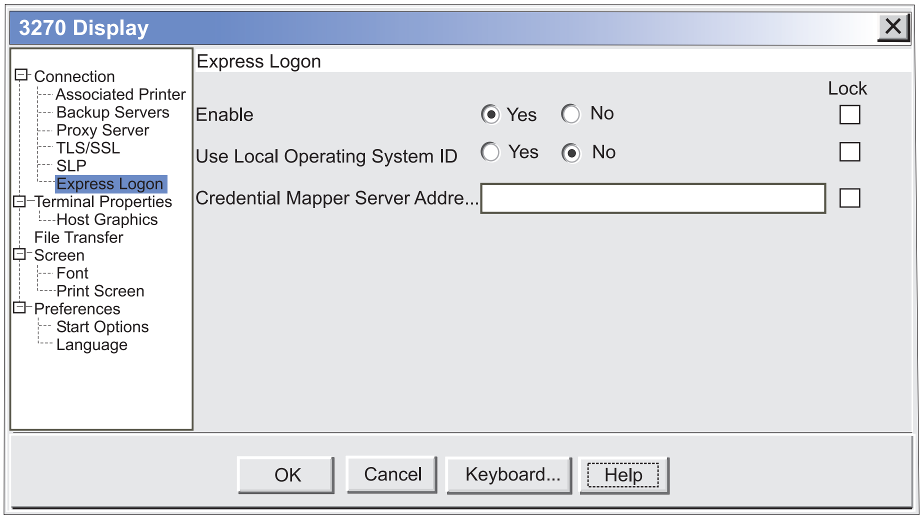
6. Add the DCAS server's certificate to the key database. Be sure that the key database content is for the signer certificate. If it is not, select the pull-down menu and change it. Then select Add.
7. Select Binary DER data for the data type. If the server certificate is in ASCII format, select Base64-encoded ASCII data.
8. Type the file name in the For Certificate File Name field.
9. Type the path name in the Location field.
10. Click OK.
11. Enter a label for the certificate and click OK.
12. Add the DCAS client's certificate to the key database.
13. Change the Key database content to Personal Certificates and click Export/Import.
14. Select Import Key as the Action Type.
15. Select PKCS12 for the Key file type.
16. Type the client certificate's p12 file name in the File Name field and path name in the Location field.
17. Click OK and enter the client certificate PIN.
18. Click OK.
19. Exit the Certificate Management GUI.

Step 4: Use the Deployment Wizard to create the HTML file.

AC Gas, Inc. uses the steps listed in "Step 4: Use the Deployment Wizard to create your HTML file." on page 27 to create an HTML file using the Deployment Wizard. They choose the HTML-based configuration model.

Step 5: Configure the Host On-Demand session.

Since AC Gas, Inc. has already installed Host On-Demand Version 8, they are ready to configure their session properties to use Web Express Logon. In the Deployment Wizard on the Host Sessions window, they highlight their 3270 session and select Properties under the Configure drop-down menu. On the left side of the window, they select Express Logon under Connection.



Enable ● Yes ○ No ☐
Use Local Operating System ID ○ Yes ● No ☐
Credential Mapper Server Addre... ☐

Figure 16. 3270 Express Logon

Once they select Yes to enable Express Logon, they type the full URL of the Credential Mapper Server, for example, https://server_name/junction/wel/CredMapper, where

- *server_name* is the name of the authentication server
- *junction* is the name of the Tivoli Access Manager junction
- *wel* is the credential mapper servlet space
- *CredMapper* is the servlet name

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.

Step 6: Record the macro.

AC Gas, Inc. records their login macro using the steps in "Step 6: Record the macro." on page 30.

At this point, AC Gas, Inc. has implemented Web Express Logon, and its employees are able to access the host system without providing additional credentials.

Part 6. API programming guide

Chapter 7. Customizing Web Express Logon

The Credential Mapper Servlet (CMS) is the core of the credential-mapping framework and resides on a Web server and some Web application framework. At a high level, it has two primary roles: (1) request the client's credentials that are supplied by the network security application and (2) respond with the host access credentials. It accomplishes these tasks through credential mapper Java objects called plug-ins. Web Express Logon provides a CMS and 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.

Depending on your environment, you can take one of two approaches for customizing the CMS. First, you may wish to replace the entire CMS with your own custom version of the servlet. In this case, you will need to use an HTTP parameter for requests and XML data for responses. Second, you may wish to use the existing CMS and integrate components using the APIs provided with Host On-Demand. This second approach requires less J2EE experience and is easier to implement.

Writing a custom Credential Mapper Servlet

Writing a custom CMS requires some J2EE knowledge and experience working with J2EE-compliant servlets.

Parameters are supplied to the CMS servlet via an HTTP request. The response information is encapsulated into an XML-formatted object and returned to the caller.

HTTP request parameters

When users first log on to their workstations or attempt to access some network resource, the network security application prompts them for their credentials. Once they are authenticated by the network security application, an HTTP request is made to the CMS using an HTTPS connection. Clients may make one or two requests to the CMS, depending on the model type of HTML page. The first request is sent to the CMS only if the HTML page was created using the Configuration server-based model. The second request is sent for sessions that are configured for macro-based automation. The Web server and application server listen for and receive the incoming request and pass it on to CMS for processing. Since it must be an HTTP request, the CMS request interface is built around a standard HTTPS 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.ibm.com/somehost&appid=tpf
  &authtype=AuthType_3270Host
```

Table 5 is a list of available keys:

Table 5. Available keys and values

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 authentication credentials being requested (available authentication types are defined below).
`localid`	This optional value will supply the user's identification, based on the local operating system. For now, this solution supported only on the Windows operating system.

Custom Credential Mapper Servlet 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. Table 6 contains the status code definitions:

Table 6. Status code definitions

Status code	Description
0	Success
1	Unknown status code
2	Credential Mapper not found
3	Invalid 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

Creating a Credential Mapper Servlet

Describing how to create a servlet is not within the scope of this document, but there are resources available to help you, for example:

- **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/.

Creating custom plug-ins for the Credential Mapper Servlet

As discussed earlier, the CMS relies on plug-ins to provide the network user ID and host access credentials. The CMS interacts with these plug-ins via Java interfaces, which are described below.

Java interfaces

All plug-ins must implement the following three Java interfaces:

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 (point user to discussion of web.xml configuration). The id parameter is the symbolic name of the plug-in specified in the CMS configuration portion of the web.xm filel. This value may be used to qualify the instance of the plug-in in the event 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.

com.ibm.eNetwork.security.sso.CMRequest

The CMRequest object is used by CMS to encapsulate all necessary parameters for a plug-in request. The following are its members and methods:

Members:

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

Methods

`public CMRequest()`

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

`public String getID()`

`public void setID(String id)`

`public String getHostApplID()`

`public void setHostApplID(String applID)`

`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()`

com.ibm.eNetwork.security.sso.CMResponse

The CMResponse object encapulates all relevant information needed by the CMS for the request made of a plug-in. The following are its members and methods:

Members:

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

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 a 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 from the incoming HTTP request object. The specifics of how to acquire the network ID is specific to your network security application. Refer to your network security documentation for more information.

Writing a Host Credential 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 Host Credential 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 during this process 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 Host Credential plug-in. In our DCAS/JDBC plug-in, we automate z/OS logins by associating a users' network IDs to their host IDs, and taking the host ID with the application ID and obtaining 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 our DCAS API. This API provides access to RACF—generated passtickets.

The DCAS API object (DCASClient) encapsulates the Passticket requests. Here are its members and methods:

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

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.

Part 7. Troubleshooting error messages

Chapter 8. Troubleshooting Web Express Logon

Web Express Logon depends on a number of independent processes working together to function properly. Some of these processes run on the Host On-Demand client while others run on other host systems. When 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. This portion of the document is devoted to that purpose.

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.

Assuming that Web Express Logon is not functioning properly (that is, you are not logged in a host emulation session), ask yourself the following questions:

1. Did the Host On-Demand client display an error message panel?
   - If yes, skip to “Web Express Logon client-side messages” on page 64.
   - If no, verify the following on your session configuration panel:
     - 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.
     - 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.
2. 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.
3. 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” on page 67.
4. Are you using IBM WebSphere Application Server and have Java 2 security enabled? If so, please 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 contain 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 Chapter 7, “Customizing Web Express Logon,” on page 53 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 Chapter 2, “Introduction,” on page 7 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 Chapter 7, “Customizing Web Express Logon,” on page 53 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 Chapter 7, “Customizing Web Express Logon,” on page 53 for more information about writing your own credential mapper 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 *CredentialMapperName* **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, is 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_3270Host,*</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_3270Host,*</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 Table 2 on page 22.

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.

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.

Appendix A. Password encryption tool

Host On-Demand provides a password encryption tool so you can encrypt your passwords for added security. The tool 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 Host Credential 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" on page 54 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 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_directory* 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.

Appendix B. Glossary of terms

The following terms are used throughout this document:

authentication type

When editing Credential Mapper Servlet (CMS)-related parameters in the web.xml file for macro-based automation, the parameter value must contain the full class path name of the implementing class, the authentication type to be addressed by the credential mapper, and the host mask.

Once you specify the desired authentication type, the CMS can better identify which credential mapper to select to handle the request. You can pair multiple authentication types together to give Host Credential Mappers (HCM) the freedom to support multiple authentication types.

connection-based automation

Connection-based automation works in iSeries environments that support Kerberos network authentication. With this type of automation, the user is authenticated through a telnet negotiation and the host never sends a login screen to authenticate the client. Therefore, connection-based automation does not require the use of a login macro, the Credential Mapper Servlet (CMS), a Network Security plug-in, nor the Host Credential Mapper (HCM).

credential challenges

Credential challenges are the time at which users are prompted to provide IDs and passwords.

Credential Mapper Servlet (CMS)

For the macro-based automation style of Web Express Logon, the CMS is the core of the credential-mapping framework. It is supplied with Host On-Demand and must be deployed to a Web server or some type of Web application framework. At a high level, it has two primary roles: (1) request the client's credentials (called a *network ID*) and (2) respond with the host access credentials, which consist of the *host ID* and a password or passticket, depending on the type of HCM.

Digital Certificate Access Server (DCAS)

DCAS is a TCP/IP server that runs on z/OS and OS/390 platforms. TN3270 servers connect to DCAS using Secure Socket Layer (SSL). The purpose of DCAS is to receive an application ID and a digital certificate from a TN3270 server and then ask RACF to return a valid user ID that has been associated with the certificate and to generate a passticket for the input user ID and application ID.

Enterprise Identity Mapping (EIM)

EIM is an IBM eServer technology that helps you easily manage multiple user registries and user identities in an enterprise. EIM is an architecture for describing the relationships between individuals or entities (like file servers and print servers) in the enterprise and the many identities that represent them within an enterprise. In addition, EIM provides a set of APIs that allow applications to ask questions about these relationships.

full class path name

When editing CMS-related parameters in the web.xml file for macro-based automation, the parameter value must contain the full class path name of the implementing class, the authentication type to be addressed by the credential mapper, and the host mask.

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 network security or HCM queries. You must place the specified class file 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.

Host Credential Mapper (HCM)

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 requires you to write your own plug-in..

host ID

A host ID is the credential used to uniquely identify the user to the host being accessed. In macro-based automation, the host ID is what the Host Credential Mapper returns to the Credential Mapper servlet in order to achieve single sign-on.

host mask

When editing CMS-related parameters in the web.xml file for macro-based automation, the parameter value must contain the full class path name of the implementing class, the authentication type to be addressed by the credential mapper, and the host mask.

The host mask is a secondary selection criteria used by the CMS to identify the most appropriate credential mapper. This value can contain one or more host addresses.

Kerberos

Kerberos is a network authentication protocol that identifies and authenticates users who request to log on to a network. It provides a means of verifying the identities of principals (users) on physically insecure networks. It provides mutual authentication, data integrity, and privacy under the realistic assumption that network traffic is vulnerable to capture, examination, and substitution.

macro-based automation

Macro-based automation is for environments of varying host types that *are not* using Kerberos authentication. As the name implies, it requires you to create a macro to perform logon automation.

In order to use the macro-based automation style of Web Express Logon, you must have a network security application in place. Host On-Demand provides out-of-the-box support for three common network security applications without requiring additional coding: IBM Tivoli Access Manager, Netegrity Siteminder, and 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.

network ID

A network ID is the credential that uniquely identifies the user to the network security application. In macro-based automation, 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

In macro-based automation, the Network Security plug-in acquires the user's network ID from the network security application.

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.

Appendix C. Sources for more information

Use the following sources to help you implement Web Express Logon in your environment:

- Host On-Demand: http://www.ibm.com/software/webservers/hostondemand
- Host On-Demand InfoCenter: http://www.ibm.com/software/webservers/hostondemand/library/v8infocenter/
- IBM redbooks: http://www.redbooks.com
- IBM WebSphere Application Server: http://www.ibm.com/software/websphere/appserv
- IBM eServer iSeries Enterprise Identity Mapping document: http://publib.boulder.ibm.com/iseries/v5r2/ic2924/info/rzalv/rzalv.pdf
- Enterprise Identity Mapping Web site: http://www.ibm.com/servers/eserver/security/eim/
- IBM eServer iSeries Resource Library: http://www.ibm.com/eserver/iseries
- IBM Tivoli Access Manager: http://www.ibm.com/software/tivoli/products/access-mgr-e-bus/
- IBM Tivoli Identity Manager: http://www.ibm.com/software/tivoli/products/identity-mgr/
- IBM DB2 Universal Database: http://www.ibm.com/software/data/db2/udb/
- OS/400 Version 5R2: http://www.ibm.com/servers/eserver/iseries/software/v5r2.html
- RACF homepage: http://www.ibm.com/servers/eserver/zseries/zos/racf/

Appendix D. Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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.

Appendix E. Trademarks

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.

Printed in USA

SC31-6377-00