IBM WebSphere Host On-Demand Version 8.0

Planning, Installing, and Configuring Host On-Demand

SC31-6301-02

IBM WebSphere Host On-Demand Version 8.0

Planning, Installing, and Configuring Host On-Demand

SC31-6301-02

Note
Before using this information and the product it supports, read the information in Appendix E, "Notices", on page 185.

Fifth 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

About this book

The *Planning, Installing, and Configuring Host On-Demand* guide (which replaces the Host On-Demand *Getting Started* guide) helps you to plan for, install, and configure the Host On-Demand program. This book is written for administrators. It contains three major parts.

Part 1, "Planning for Host On-Demand", on page 1 gives you information about Host On-Demand for you to consider before installation and deployment. For example, which server platform will you use? Do you want to take advantage of any Java 2 functions? Which deployment model will you use? How will you handle security?

Part 2, "Installing, upgrading, and uninstalling Host On-Demand", on page 59 offers step-by-step procedures based on each operating system.

Part 3, "Configuring Host On-Demand", on page 81 describes different configuration models to specify how session configuration information is defined and managed, how to dynamically modify session configuration information, how to customize new clients, and how to deploy Host On-Demand to your users.

After you install and configure Host On-Demand, use the Online Help to learn how to define sessions and perform other administrative tasks.

Planning, Installing, and Configuring Host On-Demand is also available on the CD-ROM and the Host On-Demand Web InfoCenter at http://www.ibm.com/software/webservers/hostondemand/library/v8infocenter/.

About the other Host On-Demand documentation

In addition to the *Planning, Installing, and Configuring Host On-Demand* guide, Host On-Demand also provides other sources of information to help you use the product. To access the documentation described here, go to the Host On-Demand Web InfoCenter at http://www.ibm.com/software/webservers/hostondemand/library/v8infocenter/. Most of the documentation is also included on the Host On-Demand product or Toolkit CD-ROMs.

- *Online Help*. The Online Help is the primary source of information for administrators and users after Host On-Demand installation is complete. It provides detailed steps on how to perform Host On-Demand tasks. A table of contents and an index help you locate task-oriented help panels and conceptual help panels. While you use the Host On-Demand graphical user interface (GUI), help buttons bring up panel-level help panels for the GUI.
- *Program Directory*. The program directory instructs you on how to install Host On-Demand on the z/OS and OS/390 platforms.
- *Readme file*. This file, readme.html, contains product information that was discovered too late to include in the product documentation.
- *Web Express Logon Reference*. This book provides a step-by-step approach for understanding, implementing, and troubleshooting Web Express Logon. It offers an overview of Web Express Logon, two scenario-based examples to help you plan for and deploy Web Express Logon in your own environment, as well as several APIs for writing customized macros and plug-ins.

- *Macro Programming Guide*. This book describes how to create Host On-Demand macros for automating user interactions with a 3270 or a 5250 host application or for passing data between a host application and a native application. This book provides detailed information on all aspects of developing macros and includes revised information about the macro language previously published in the Host Access Beans for Java Reference.
- *Host Printing Reference*. After you configure host sessions, use the Host Printing Reference to enable your users to print their host session information to a local or LAN-attached printer or file.
- *Session Manager API Reference*. This book provides JavaScript APIs for managing host sessions and text-based interactions with host sessions.
- *Programmable Host On-Demand*. This book provides a set of Java APIs that allows developers to integrate various pieces of the Host On-Demand client code, such as terminals, menus, and toolbars, into their own custom Java applications and applets.
- *Toolkit Getting Started*. This book explains how to install and configure the Host On-Demand Toolkit, which is shipped with the Host Access Client Package, but is installed from a different CD-ROM than the Host On-Demand base product. The Host On-Demand Toolkit complements the Host On-Demand base product by offering Java beans and other components to help you maximize the use of Host On-Demand in your environment.
- *Host Access Beans for Java Reference*. This book is part of the Host On-Demand Toolkit. It serves as a reference for programmers who want to customize the Host On-Demand environment using Java beans and create macros to automate steps in emulator sessions.
- *Host Access Class Library Reference*. This book is part of the Host On-Demand Toolkit. It serves as a reference for programmers who want to write Java applets and applications that can access host information at the data stream level.
- *J2EE Connector Reference*. This book is part of the Host On-Demand Toolkit. It serves as a reference for programmers who want to write applets and servlets that access Java 2 Enterprise Edition (J2EE) compatible applications.
- *Host On-Demand Redbooks*. The Host On-Demand Redbooks complement the Host On-Demand product documentation by offering a practical, hands-on approach to using Host On-Demand. Redbooks are offered "as is" and do not always contain the very latest product information. For the most up-to-date list of all Host On-Demand Redbooks, visit the Host On-Demand library page at http://www.ibm.com/software/webservers/hostondemand/library.html.

Conventions used in this book

The following typographic conventions are used in *Planning, Installing and Configuring Host On-Demand*:

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.
Return	Refers to the key labeled with the word Return, the word Enter, or the left arrow.

Table 1. Conventions used in this book (continued)

Convention	Meaning
>	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.

Terminology

This section describes the terminology used throughout this book.

applet A small application program that performs a specific task and is usually portable between operating systems. Often written in Java, applets can be downloaded from the Internet and run in a Web browser.

application
A program or suite of programs that perform a task or specific function.

cached client
A Host On-Demand cached client is any Host On-Demand client whose components have been cached (stored locally for quick access) on the hard disk of a user's workstation.

default publish directory
The default publish directory is the subdirectory HOD in your Host On-Demand server's install directory, such as `c:\Program Files\IBM\HostOnDemand\HOD\` on Windows and `/opt/IBM/HostOnDemand` on AIX, Linux, Solaris, and HP-UX.

download client
Download clients download the necessary applet files each time users access the HTML files. Download clients are generally used in LAN-connected environments because high-speed network connections reduce the time it takes to download them from the Web server.

emulator client
An emulator client is a Host On-Demand client that launches a terminal emulator session. Host On-Demand includes the following emulator clients: cached client, Web Start client, and download client.

separate user publish directory
Provides a separate writeable location for deploying custom HTML files, isolating them from the files provided by Host On-Demand. This keeps the Host On-Demand publish directory read-only and makes it easier to apply

future Host On-Demand upgrades. Note that other user-modified files (such as customer applets and HACL programs) still need to run from the Host On-Demand publish directory.

Web Application Server
The run time for dynamic Web applications. Web application server includes support for Java servlets, JavaServer Pages (JSP), and other enterprise Java application programming interfaces (APIs). A Web application server provides communications, resource management, security, transaction management, and persistence capabilities for Web applications. It also typically includes an administration interface for managing the server and deployed applications.

Web server
A server on the Web that serves requests for HTTP documents. The Web server controls the flow of transactions to and from WebSphere Commerce. It protects the confidentiality of customer transactions and ensures that the user's identity is securely transmitted to the WebSphere Commerce Server.

Web Start Client
The Web Start client allows users to run Host On-Demand sessions without a browser. Users start Host On-Demand sessions from the Java Web Start Application Manager.

Terms relating to Java 1 and Java 2

Note the following terms and their use in this document.

Java 1 Refers to a Java 1.1.x Java Virtual Machine (JVM).

Java 2 Refers to a Java 1.3.x, 1.4.x, or later JVM.

Java 1 browser
A Web browser that runs Java applets on a Java 1 JVM that is usually included with the browser, for example, Netscape 4.7x and Internet Explorer without a Java 2 plug-in. For more information, refer to "Browsers and Java 2 plug-ins" on page 31.

Java 2-enabled browser
A Web browser that runs Java applets on the Java 2 JVM of an installed Java 2 plug-in, for example, Netscape 7.0 and Internet Explorer with a Java 2 plug-in. For more information, refer to "Browsers and Java 2 plug-ins" on page 31.

Java 1 emulator client, Java 1 cached client, Java 1 download client
A version of the Host On-Demand client. The Java 1 version consists of a complete set of Host On-Demand client components compiled with a Java 1 compiler. For more information, refer to "Java 1 and Java 2 versions of the Host On-Demand emulator client" on page 31.

Java 2 emulator client, Java 2 cached client, Java 2 download client
A version of the Host On-Demand client. The Java 2 version consists of a complete set of Host On-Demand client components compiled with a Java 2 compiler. For more information, refer to "Java 1 and Java 2 versions of the Host On-Demand emulator client" on page 31.

Part 1. Planning for Host On-Demand

Chapter 1. Introducing WebSphere Host On-Demand

What is WebSphere Host On-Demand?

IBM WebSphere Host On-Demand provides cost effective and secure browser-based and non-browser-based host access to users in intranet-based and extranet-based environments. Host On-Demand is installed on a Web server, simplifying administrative management and deployment, and the Host On-Demand applet or application is downloaded to the client browser or workstation, providing user connectivity to critical host applications and data.

Host On-Demand supports emulation for common terminal types, communications protocols, communications gateways, and printers, including the following:

- TN3270 and TN3270E terminals
- TN5250 terminals
- VT52, VT100, VT220, VT320, and VT420 terminals
- The Secure Shell (SSH)
- File Transfer Protocol (FTP)
- Customer Information and Control System (CICS) Transaction Gateway
- TN3270E and TN5250 printers

You can use the Java component-based Host Access Toolkit to create customized e-business applications. This Toolkit contains a rich set of Java libraries and application programming interfaces: Host Access Class Library (HACL), Host Access Beans for Java, and Java 2 Enterprise Edition (J2EE) connectors. Host On-Demand also includes Database On-Demand, which provides an interface for sending Structured Query Language (SQL) queries to IBM DB2 databases hosted on iSeries systems.

How does Host On-Demand work?

The following figure and explanation show how a Host On-Demand system works. Host On-Demand is a client/server system. Host On-Demand clients are Java applets that are downloaded from the Web server to a Web browser on a remote computer.

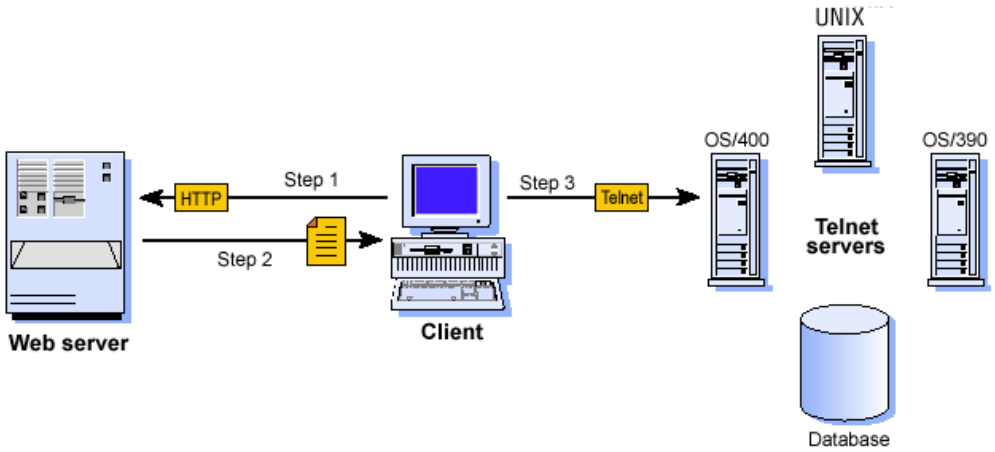

Figure 1. How Host On-Demand works

Step 1. The user opens a browser and clicks a hyperlink.

Step 2. IBM WebSphere Host On-Demand applet downloads to the client workstation.

Step 3. When the applet is downloaded, IBM WebSphere Host On-Demand connects directly to any Telnet server to access host applications.

Session information is configured in the HTML file or Host On-Demand configuration server. For more information about the configuration server, see Chapter 3, “Planning for deployment”, on page 23.

Host On-Demand client applets can be run as download clients, Web Start clients or cached clients.

- Download clients are downloaded from the Web server every time they are used.
- Web Start clients can be downloaded from the Host On-Demand Web server, or they can be installed from a LAN or CD drive.
- Cached clients are downloaded from the Web server and stored on the client computer. After the initial download, the cached client is loaded from the local machine. The cached client checks the Host On-Demand server for new versions of the client and automatically downloads the updated version.

Host On-Demand includes the following administrative components:

- The Deployment Wizard, a tool for creating emulator client HTML files. The Deployment Wizard enables administrators to quickly and easily build Host On-Demand HTML files that are customized for an organization’s needs.
- Administration clients that can be used by system administrators to define common sessions, create users and groups, and perform other administrative tasks on the Host On-Demand server.

In addition, a number of predefined clients are also supplied with Host On-Demand to demonstrate Host On-Demand’s client functions for users and administrators (for example, emulation, Database On-Demand, cached client removal, and problem determination utilities).

Why use Host On-Demand?

A cost-effective approach to connectivity

You can reduce maintenance costs and increase your return on investment by installing Host On-Demand on a Web server, eliminating the need to manage individual user desktops.

Since the applets reside on a server and are downloaded to Web browsers when needed, you no longer have to schedule maintenance and upgrades. Upgrade the software on the server and users can receive the upgrade the next time they access the client applet.

Centralized management of configuration data

Administrators can centrally define and control all session configuration information available to their users, including connection options, security features, macro definitions, keyboard specifications, and color mappings. Furthermore, administrators have full control over which fields the user can or cannot modify, and can choose where user updates should be stored.

Connect directly to any Telnet server

With Host On-Demand, the client applet contains the emulation functionality. This eliminates the need for a middle-tier server which resolves a performance and security issue. Once the applet is served to the client, it is easy to connect directly to any standard Telnet server that provides the best access to the required data. You can access many host sessions concurrently. By eliminating the need for a middle-tier server, Host On-Demand also minimizes capacity restrictions. To see how this works, refer to Figure 1 on page 4.

Browser-based user interface

The browser-based access of Host On-Demand gives you a simple way to centrally manage and deploy critical host applications and data. Host On-Demand uses the power of Java technology to open the doors to your host system whenever you need it, wherever you need it, directly from your browser. Just click on a hyperlink to launch the Host On-Demand Java applet. This Web-to-host connectivity solution provides secure Web-browser access to host applications and system data through Java-based emulation, so you can take existing host applications to the Web without programming. Because Host On-Demand is Java-based, its interface has the same look-and-feel across various types of operating environments.

Supports many different platforms and network environments

Host On-Demand servers and clients are supported on a wide variety of platforms and can be used over any TCP/IP network. This gives you a great deal of flexibility in setting up your system and enables Host On-Demand to be deployed in your computing environment without having to purchase new hardware.

Java 1 and Java 2 support

Host On-Demand is compatible with browsers that support either the Java 1 or Java 2 standards. In addition, some new features of Host On-Demand take advantage of capabilities offered only by Java 2.

Supports many national languages

Host On-Demand is available in 23 languages, including double-byte character set (DBCS) languages. Support for the European currency symbol, as well as keyboard and code page support for many more languages such as Arabic, Hebrew and Thai, is also provided. All language versions are available on the same media, and multiple language versions can be accessed concurrently.

Secure connections

Using Transport Layer Security (TLS) version 1.0 and Secure Sockets Layer (SSL) Version 3.0, Host On-Demand extends secure host data access across intranets, extranets, and the Internet. Mobile workers access a secure Web site, receive authentication and establish communication with a secure enterprise host. With client and server certificate support, Host On-Demand can present a digital certificate (X.509, Version 3) to the Telnet server - such as IBM Communications Server for Windows NT Version 6 or later, or IBM Communications Server for OS/390 Version 2.6 or later - for authentication.

Host On-Demand can also be configured for use in environments that include firewalls. Firewall ports need to be opened for the functions defined in your Host On-Demand session definitions. For more information, refer to "Using Host On-Demand with a firewall" on page 47.

Custom HTML files

Host On-Demand includes a Deployment Wizard that enables you to create custom HTML files. These files enable you to tailor the content of the client and the function necessary to meet the needs of specific groups of users. For more information about the Deployment Wizard, refer to Chapter 10, "Configuring Host On-Demand emulator clients", on page 83.

Toolkit for creating new e-business applications

Host On-Demand includes the Java component-based Host Access Toolkit for creating customized e-business applications. This Toolkit contains a rich set of Java libraries and application programming interfaces, including the Host Access Class Library (HACL), Host Access Beans for Java, and Java 2 Enterprise Edition (J2EE) connectors.

HACL provides a non-visual API for interacting with back-end host machines running applications originally designed for human interaction. Host applications rely on readable character presentation, formatted fields, color-coding, and keyboard responses. HACL provides specialized classes for functionalities needed to mimic traditional interaction with a series of host screen presentations (green screens). HACL contains no GUI (visible component) classes. For example, a Java program could be running on a mainframe as a middle man. The middle man program interacts first with another mainframe running a CICS data application, and then with a client browser through dynamically generated HTML pages. The middle man interprets client inputs into simulated terminal actions which are sent to the CICS machine using the HACL API. The response screens from the CICS machine are captured using HACL APIs, converted into dynamic HTML pages, and sent back to the client.

Host On-Demand J2EE Connector provides a set of Resource adapters that communicate to 3270, 5250, CICS, and VT hosts. These resource adapters are deployed to a conforming application server, such as IBM WebSphere Application

Sever. The users can write Web applications using the APIs provided in Host On-Demand J2EE Connector via WebSphere Studio Application Developer Integration Edition.

Programmable Host On-Demand

Programmable Host On-Demand is a set of Java APIs that allows developers to integrate various pieces of the Host On-Demand client code, such as terminals, menus, and toolbars, into their own custom Java applications and applets. The API gives the developer complete control over the Host On-Demand desktop (what the user sees) without starting with the Host Access Java Beans found in the Toolkit. The underlying Host On-Demand code handles all the ″wiring″ of the various components, including saving user preferences, such as macros, keyboard remappings, and color remappings, to the local file system for future use. The developer must only determine the layout of the Host On-Demand desktop. For more information, refer to the `Programmable Host On-Demand Reference` .

Host On-Demand Session Manager APIs

In addition to the application programming interfaces (APIs) provided with the Host Access Toolkit, Host On-Demand provides specialized public APIs that provide support for embedding host sessions in Web pages using JavaScript. These JavaScript-based APIs help application developers manage host sessions and text-based interactions with host sessions and are available through the Host On-Demand Session Manager. Refer to the Session Manager API Reference for more information.

Support for WebSphere Portal

Host On-Demand can run as a portlet on Portal Server, a component of WebSphere Portal. Portal Server has sophisticated desktop management and security features that offer administrators more control over user access rights and end users control over the appearance and arrangement of the portal desktop.

Administrators can create customized Host On-Demand portlets quickly and easily using the Deployment Wizard and then load them directly into Portal Server. (Note that Portal Server is a separate product and requires independent installation.)

Connections to DB2 databases on iSeries

Database On-Demand is included with Host On-Demand to provide access to DB2 information stored on iSeries computers using a Java Database Connectivity (JDBC) driver. Database On-Demand is a Java applet that allows you to perform Structured Query Language (SQL) requests to iSeries databases through a JDBC driver.

What's new?

Getting the latest information on Host On-Demand

For the most recent information on Host On-Demand Version 8, see the readme file.

For up-to-date product information, go to the Host On-Demand Web site at `http://www.ibm.com/software/webservers/hostondemand`.

For the latest technical hints and tips for Host On-Demand, go to the `Host On-Demand Hints and Tips site`.

To subscribe to the Software Support Bulletin, go to http://www.ibm.com/software/network/support.

New features in Host On-Demand Version 8

The following functions and enhancements were added to Host On-Demand Version 8:

Security

Web Express Logon: Web Express Logon allows users to access host systems and host applications without having to provide a user ID and password. Unlike Certificate Express Logon, Web Express Logon works in conjunction with your existing network security application and does not require client certificates. For more information, refer to the Web Express Logon Reference.

SSH (the Secure Shell) (Java 2 only): Host On-Demand now supports secure VT Display sessions and secure file transfer protocol (sftp) sessions using SSH. SSH is a popular protocol for running secure sessions over a non-secure transport layer such as TCP/IP. Host On-Demand supports a subset of SSH Version 2. For more information, refer to "Secure Shell (SSH)" on page 41.

Secure FTP: The Host On-Demand FTP client now provides TLS and SSL-based secure file transfer. For more information, refer to "TLS and SSL for Host On-Demand" on page 37.

Technology

Web Start (Java 2 only): Web Start client is new Java 2 technology that provides the ability to run Host On-Demand without a browser. This client type can eliminate some of the problems that occur when running Host On-Demand in a browser, such as inadvertently closing the browser when a session is active. For more information, refer to "Web Start client" on page 107.

TN3270E Functional Extensions: Host On-Demand 3270 Display sessions now support the TN3270E protocol. TN3270E is an enhanced form of the TN3270 protocol that allows users to specify an LU or LU pool to which the session will connect. TN3270E also supports the Network Virtual Terminal (NVT) protocol for connecting to servers in ASCII mode (for example, in order to log on to a firewall). For more information, refer to TN3270E in the online help.

Host On-Demand also supports the contention-resolution mode feature of TN3270E. This feature allows the 3270 host to indicate to the client when the host has finished updating the application screen. Several parameters have been added to the Host On-Demand macro language to take advantage of contention-resolution mode. For more information, refer to the Macro Programming Guide.

Support for Internet Protocol Version 6 (Java 2 only): Internet Protocol Version 6 (IPv6) is the next generation of the Internet Protocol. With the introduction of IPv6, the current version of the Internet Protocol is referred to as Internet Protocol Version 4 (IPv4). IPv6 expands the number of available Internet addresses and provides improvements over IPv4 in the areas of routing and network configuration. Java 1.4 is required for this support. For more information, refer to "Support for Internet Protocol Version 6" on page 19.

Macintosh support (Java 2 only): The Host On-Demand emulator and database clients now support Mac OS X. For more information, refer to "Apple Mac OS X with Java 2" on page 29 and "Cached client support for Mac OS X (Java 2 clients only)" on page 105.

Cached client improvements for Java 2: Host On-Demand offers several improvements in the Java 2 cached client. For most types of Java 2 cached clients, including the emulator client, the Database On-Demand client, and the new user client, you can now do the following:

- install the client from a LAN drive or CD
- share the client between more than one user on Windows 2000 or Windows XP
- upgrade the client in the background
- remove the client in one operation, without having to clear the Java 2 plug-in's cache

These improvements raise the Java 2 cached client to the same level of user-friendliness and flexibility as the Java 1 cached client. For more information, refer to "Improvements to the cached client for Java 2" on page 28.

Programmable Host On-Demand: Programmable Host On-Demand allows you to incorporate the Host On-Demand terminal and its GUI elements into your own application without having to wire the beans. For more information, refer to `Programmable Host On-Demand Reference`.

Unicode support for OS/400: For 5250 Display sessions, Host On-Demand now supports iSeries hosts that send Unicode data using tagged CCSID fields. This capability is supported by OS/400 V5R2. This enhancement allows Unicode data to be displayed in 5250 Display sessions. Host On-Demand supports CCSIDs 13488 (0x34B0) and 17584 (0x44B0). For more information, refer to Unicode support for OS/400 using Code Character Set Identifiers in the online help.

Productivity

ZipPrint: Host On-Demand now allows you to automatically print some types of 3270 documents in their entirety by making a single selection. This ZipPrint capability is similar to the ZipPrint functions in the IBM Personal Communications product. You can also add your own customized document types. For more information, refer to ZipPrint in the online help.

Sharing and reusing macro, keyboard, and toolbar components: Administrators and users can now share and reuse the following configuration components from one session to the next:

- macros
- keyboard definitions
- toolbar definitions

This increases productivity because administrators no longer need to reconfigure these components for each individual session or HTML file, and users can share a single component definition across sessions. For more information about sharing and reusing configuration components, refer to `Sharing and reusing macro, keyboard, and toolbar definitions` in the online help.

Macro Programming Guide: The new `Host On-Demand Macro Programming Guide`, available in both HTML and PDF format, describes how to use the Macro Editor, the Code Editor, and the macro language, and provides examples.

Macro enhancements: Host On-Demand offers new enhancements to the macro capability:

- You can invoke methods in Java classes. For more information see the topic `Imported types` in the online help.
- You can print screens or a series of screens using new print actions.

Server macro libraries: The Host Sessions window of the Deployment Wizard now allows you to deploy server macro libraries to selected sessions from either a Web server or a shared network drive. This eliminates the need to import large macros into a session, and you can update them without modifying the Host On-Demand sessions or HTML definitions.

For more information about server macro libraries, refer to Chapter 14, "Creating and deploying server macro libraries", on page 121 in this guide.

Redirector enhancements: The Redirector now provides connection logging and allows connections to remain active during a period of inactivity. You can specify the amount of time to wait before dropping an inactive Redirector connection. For more information, refer to `Adding a host to the Redirector` in the online help.

List function in the Directory Utility: Directory Utility now adds a list function to its other actions, all calculated to save time by allowing management of configuration data in a batch mode environment. With this new function you can perform searches on users and groups, using criteria with flexible wildcard options. The list function writes search results to output files that can be reused as input for other actions. For more information, refer to `Using the Directory Utility` in the online help.

Usability

Backup Servers: On most session types you can specify up to two backup servers in addition to the main server. If you specify backup servers, then the Host On-Demand client automatically tries to connect with the backup servers if the connection with the main server fails. For more information, refer to `Backup Servers` in the online help.

Accessibility improvements: Based on Section 508 of the US Rehabilitation Act, Host On-Demand offers new accessibility features to help users who have physical disabilities, such as restricted mobility, limited or no vision, or limited or no hearing, use host sessions successfully. While we do not offer every accessibility feature, we do offer the following enhanced features:

- A textual version of the Operator Information Area (OIA).
- Keyboard equivalents for actions (mouseless operation).
- Support for display system settings for size, font, color, and high contrast.
- An improved Color Remap window for the emulator screen

Accessibility features require Java 1.4, which is a specific version of the Java 2 platform. For more information, refer to `Accessibility` in the online help.

CICS client improvements: Now users have more power to customize CICS Gateway sessions according to their work habits. They either can specify which CICS transaction starts upon host connection, or choose to begin their sessions without an initial transaction. For more information, refer to `CICS Initial Transaction` in the online help.

Importing and exporting sessions: The Host Sessions window of the Deployment Wizard now gives you the option to import and export entire host sessions. In previous releases of Host On-Demand, this option was only available from the Administration Utility.

For more information, refer to `Importing sessions` and `Exporting sessions` in the online help.

Chapter 2. Requirements

For updates to this information, refer to the Readme.

Server requirements

z/OS and OS/390 operating systems

For a complete list of z/OS and OS/390 requirements, see the Program Directory.

OS/400 operating systems

Table 2. Server requirements for Host On-Demand on OS/400 operating systems

Server operating system	• OS/400 V5R1 • OS/400 V5R2 Recent cumulative service is recommended. Refer to the iSeries Support, Recommended fixes Web site for service information. Unicode support using Coded Character Set Identifiers (CCSIDs) requires V5R2 with the following PTFs: • SI08903 • SI08904 • SI08933 • SI08985
Disk space	363 MB for an English-only installation. Add 4 to 8 MB for each additional national language to be installed.
Memory	256 MB memory or more. Refer to the iSeries Performance Capabilities Reference Web page for additional information about the impact of additional memory and Java performance
Supported Web servers	• Apache-based HTTP Server for iSeries • IBM HTTP Server for iSeries • Lotus Domino for iSeries (manual configuration required)
Supported Web Application Servers	• WebSphere Application Server 4.0, 5.0, and 5.0 Express • Lotus Domino for iSeries (manual configuration of servlet required)
Java	Toolbox for Java Java Developer's Kit *BASE option and one of the following: • Option 4 - 1.1.8 • Option 5 - 1.3 • Option 6 - 1.4

Table 2. Server requirements for Host On-Demand on OS/400 operating systems (continued)

All other requirements	TCP/IP Connectivity Utilities for iSeries QShell Interpreter

Windows operating systems

Table 3. Server requirements for Host On-Demand on Windows operating systems

Server operating systems	• Windows NT 4.0 with SP5 or later • Windows 2000 Professional, Server, and Advanced Server • Windows XP Professional (32-bit) • Windows Server 2003
Disk space	363 MB for an English-only installation. Add 4 to 8 MB for each additional national language to be installed.
Supported Web servers	• Apache HTTP Server V1.3 and V2.0 • IBM HTTP Server V1.3.12.6, V1.3.19.2, V1.3.26, and V2.0.42 • iPlanet Web Server Enterprise Edition V6.0 • Lotus Domino R6 (manual configuration required) • Lotus Go V4.6 • Microsoft IIS 4, 5, 5.1, and 6
Supported Web Application Servers	• iPlanet Application Server V6.0 (manual configuration of servlet required) • Lotus Domino R6 (manual configuration of servlet required) • WebSphere Application Server 4.0, 5.0, and 5.0 Express
Java	Installed with Host On-Demand

AIX operating systems

Table 4. Server requirements for Host On-Demand on AIX operating systems

Server operating system	• AIX (R) Version 4.3.3 • 5L 5.1 • 5.2
Disk space (installp image)	363 MB for an English-only installation. Add 4 to 8 MB for each additional national language to be installed (including the additional security files).

Table 4. Server requirements for Host On-Demand on AIX operating systems (continued)

Supported Web servers	• Apache HTTP Server V1.3 and V2.0 • IBM HTTP Server V1.3.12.6, V1.3.19.2, V1.3.26, and V2.0.42 • iPlanet Web Server Enterprise Edition V6.0 • Lotus Domino R6 (manual configuration required) • Lotus Go V4.6
Supported Web Application Servers	• iPlanet Application Server V6.0 (manual configuration of servlet required) • Lotus Domino R6 (manual configuration of servlet required) • WebSphere Application Server 4.0 and 5.0
C/C++ Runtime Libraries	• AIX Version 4.3.x requires level 5.0.2.0 • AIX Version 5.x requires level 6.0.0.3 C/C++ runtime libraries are available for download at ftp://www7b.boulder.ibm.com/aix/fixes/byCompID/5765F5600/.
Java	Installed with Host On-Demand

Solaris operating systems

Table 5. Server requirements for Host On-Demand on Solaris operating systems

Server operating system	• 7 • 8 • 9
Disk space	363 MB for an English-only installation. Add 4 to 8 MB for each additional national language to be installed.
Supported Web servers	• Apache HTTP Server V1.3 and V2.0 • IBM HTTP Server V1.3.12.6, V1.3.19.2, V1.3.26, and V2.0.42 • iPlanet Web Server Enterprise Edition V6.0 • Lotus Domino R6 (manual configuration required) • Lotus Go V4.6
Supported Web Application Servers	• iPlanet Application Server V6.0 (manual configuration of servlet required) • Lotus Domino R6 (manual configuration of servlet required) • WebSphere Application Server 4.0 and 5.0
Java	Installed with Host On-Demand

HP-UX operating systems

Table 6. Server requirements for Host On-Demand on HP-UX operating systems

Server operating system	• 11.0 • 11i
Disk space	363 MB for an English-only installation. Add 4 to 8 MB for each additional national language to be installed.
Supported Web servers	• Apache HTTP Server V1.3 and V2.0 • IBM HTTP Server V1.3.12.6, V1.3.19.2, V1.3.26, and V2.0.42 • iPlanet Web Server Enterprise Edition V6.0 • Lotus Domino R6 (manual configuration required) • Lotus Go V4.6
Supported Web Application Servers	• iPlanet Application Server V6.0 (manual configuration of servlet required) • Lotus Domino R6 (manual configuration of servlet required) • WebSphere Application Server 4.0 and 5.0
Java	Installed with Host On-Demand

Linux operating systems

Table 7. Server requirements for Host On-Demand on Linux operating systems

Server operating systems	• Red Hat Linux 7.1, 7.2, 7.3, Red Hat Enterprise Linux AS 2.1, 8.0 Personal, 8.0 Professional, 9.0 Personal, and 9.0 Professional • SuSE Linux 7.1, 7.2, 7.3, 8.0, 8.1 Professional, and 8.2 • Caldera 3.1, SCO-Caldera OpenLinux Workstation 3.1.1, and SCO-Caldera OpenLinux Server 3.1.1 • TurboLinux 6.5, 7.0, 8.0 Workstation, and 8.0 Server
Disk space	363 MB for an English-only installation. Add 4 to 8 MB for each additional national language to be installed.
Supported Web servers	• Apache HTTP Server V1.3 and V2.0 • IBM HTTP Server V1.3.12.6, V1.3.19.2, V1.3.26, and V2.0.42 • iPlanet Web Server Enterprise Edition V6.0 • Lotus Domino R6 (manual configuration required) • Lotus Go V4.6

Table 7. Server requirements for Host On-Demand on Linux operating systems (continued)

Supported Web Application Servers	• iPlanet Application Server V6.0 (manual configuration of servlet required) • Lotus Domino R6 (manual configuration of servlet required) • WebSphere Application Server 4.0, 5.0, and 5.0 Express
Java	Installed with Host On-Demand

OS/2 operating systems

Table 8. Server requirements for Host On-Demand on OS/2 operating systems

Server operating system	• OS/2 (R) Warp Server Version 4 • OS/2 Warp Server for e-Business 4.5
Disk space	510 MB. The hard disk must be configured for HPFS.
Supported Web servers	Lotus Domino Go Web server for OS/2
Java	OS/2 JDK 1.1.8 or JDK 1.3.

You can obtain the latest OS/2 JDK from one of the following Web sites:

```
ftp://ftp.hursley.ibm.com/pub/java/
http://www.ibm.com/java
```

For JDK 1.1.8, make sure your classpath entry in `config.sys` is updated with the location of the JDK class files and that the current directory (.) is included. The classpath should include something like this:

```
c:\Java11\lib\classes.zip;
```

When you have installed the JDK and set the classpath, reboot the workstation so that the updated classpath takes effect.

Novell Netware operating systems

Table 9. Server requirements for Host On-Demand on Novell Netware operating systems

Server operating system	• 4.2 • 5.1 • 6
Disk space	510 MB
Supported Web servers	Novell Web Server
Java	Novell JDK 1.1.8 and JDK 1.3

You can obtain the latest Novell JDK at `http://www.developer.novell.com`. The JDK must be configured for long-filename support.

For users to load the client HTML files from a Novell server, their browsers might need to be configured not to use a proxy server. In addition, if users have a browser with a Java 2 plug-in, the IBM plug-in must be 1.3.1 or later and the Sun plug-in must be version 1.3.1 or later. The client applets do not successfully load if the plug-in is an earlier version.

LDAP servers

The Host On-Demand server can optionally use the lightweight directory access protocol (LDAP) as a data store for user and group information. The following LDAP servers are supported:

- IBM LDAP Directory Server V3.2.2
- IBM Directory Server V4.1 and V5.1
- IBM LDAP Server running on OS/390 V2R9 and V2R10
- IBM LDAP Server running on z/OS V1R1, V1R2, V1R3, and V1R4
- Netscape Directory Server V4.0 (Windows NT and AIX)

For more information on IBM's LDAP Directory solution and to download a complimentary evaluation kit, go to
`http://www.software.ibm.com/network/directory/`

For instructions on using LDAP with Host On-Demand, see Chapter 19, "Configuring Host On-Demand Server to use LDAP", on page 161.

Web servers

Host On-Demand supports the following Web servers:

- Lotus Domino R6
- Lotus Go V4.6
- iPlanet Web Server Enterprise Edition V6.0
- IBM HTTP Server V1.3.12.6, V1.3.19.2, V1.3.26, V2.0.42
- Apache HTTP Server V1.3 and V2.0
- Microsoft IIS 4, 5, 5.1, and 6

Web Application Servers

Host On-Demand supports the following Web Application Servers:

- WebSphere Application Server 4.0, 5.0, and 5.0 Express
- iPlanet Application Server V6.0
- Lotus Domino R6

Development Environments

Host On-Demand supports the following Development Environments:

- IBM's VisualAge for Java Version 3.5 and 4.0
- WebSphere Studio Application Developer 4.0 and 5.0
- WebSphere Studio Site Developer Advanced 4.0
- Borland/Inprise's JBuilder Version 5.0, 6.0, 7.0, and 8.0

Miscellaneous software

- IBM WebSphere Portal for Multiplatforms 2.1, 4.1, and 4.2
- CICS Transaction Gateway 5.0.1

- Acrobat Reader (Acrobat) Version 4.0 or later (Note: Acrobat Version 5.0 or later is required for DBCS PDF support.)
- Netegrity Siteminder 5.5
- Tivoli Access Manager for e-business 4.1

Support for Internet Protocol Version 6

Internet Protocol Version 6 (IPv6) is the next generation of the Internet Protocol designed by the Internet Engineering Task Force (IETF) to replace the current version, which is now referred to as Internet Protocol Version 4 (IPv4). For almost 20 years, IPv4 has been very effective for the Internet, but now it is experiencing certain limitations. For example, the main limitation is the growing shortage of IPv4 addresses, which is a big concern as the Internet continues to grow with more and more machines needing IP addresses. IPv6 expands the number of available IP addresses and makes improvements in areas such as routing and network configuration. IPv6 is expected to gradually replace IPv4, with the two coexisting for a number of years during a period of transition.

Support for IPv6 requires Java 1.4.

Host On-Demand 8 supports IPv6 on the following operating systems:

- Solaris Version 8 and later
- Linux kernel 2.1.2 and later (Red Hat Version 8 and later)

For an up-to-date list of operating systems supported by Host On-Demand that use IPv6, visit the Host On-Demand V8 Web InfoCenter at: http://www.ibm.com/software/webservers/hostondemand/library/v8infocenter/.

Client requirements

For updates to client requirements, refer to the Readme, readme.html.

Supported operating systems

Host On-Demand clients are supported on the following operating systems:

- Windows 95

Host On-Demand does not support Windows 95 as a local client.

- Windows 98
- Windows NT 4.0 with SP5 or later
- Windows Millennium Edition (ME)
- Windows 2000 (Professional)
- Windows XP Professional and Home Edition (32-bit version)
- Windows Server 2003
- AIX 4.3.3, 5L 5.1, and 5.2
- OS/2 Warp 4
- Sun Solaris 7, 8, and 9
- HP-UX 11.0 and 11i
- Red Hat Linux 7.1, 7.2, 7.3, Red Hat Enterprise Linux AS 2.1, 8.0 Personal, 8.0 Professional, 9.0 Personal, and 9.0 Professional
- SuSE Linux 7.1, 7.2, 7.3, 8.0, 8.1 Professional, and 8.2

- Caldera 3.1, SCO-Caldera OpenLinux Workstation and SCO-Caldera OpenLinux Server
- TurboLinux 6.5, 7.0, 8.0 Workstation, and 8.0 Server
- Microsoft Windows NT Server 4.0 Terminal Server Edition
- Windows Terminal Services for Windows 2000
- Netstation V2R1M0
- Citrix Metaframe 1.8 for Windows Terminal Server 4.0 and 1.8 for Windows 2000 Server
- Citrix Metaframe XP Presentation Server (Versions s,a,e) for Windows
- Mac OS X 10.2.1

Host On-Demand does not support Netscape on Mac OS X.

Host On-Demand supports a local client only on Windows 98, Windows NT, Windows Millennium, Windows 2000, and Windows XP.

Supported browsers and Java 2 plug-ins

For the most up-to-date list of supported Web browsers and Java 2 plug-ins see the Readme and the Host On-Demand Web site.

The supported browsers run either a Host On-Demand local client (that is, a download client or cached client downloaded to the workstation from a Host On-Demand server, see Chapter 12, “Using Host On-Demand emulator clients”, on page 95) or a Host On-Demand locally installed client (see Appendix A, “Using locally installed clients”, on page 165).

Supported Java 1 browsers

Host On-Demand supports the following Java 1 browsers:

- Netscape Navigator 4.7

Host On-Demand does not support Netscape on Mac OS X.

- Netscape Navigator (OS/2) 4.61
- Microsoft Internet Explorer 4.01, 5.01, 5.5, and 6.0 without a Java 2 plug-in installed.

Supported Java 2–enabled browsers and Java 2 plug-ins

Host On-Demand supports the following Java 2–enabled browsers:

- Netscape Navigator 6.1, 6.2, 7.0

Host On-Demand does not support Netscape on Mac OS X.

- IBM Web Browser for OS/2 V1.2
- Microsoft Internet Explorer 4.01, 5.01, 5.5, and 6.0 with a Java 2 plug-in installed.
- Safari
- Mozilla 1.0.2, 1.2.1

A Java-2 enabled browser requires a Java 2 plug-in. Supported Java 2 plug-ins are: Sun, IBM, and HP Java plug-ins 1.3.1, 1.4.0, and 1.4.1.

For more information about Java 2–enabled browsers and Java 2 plug-ins, refer to Chapter 4, “Planning for Java 2 on the client”, on page 27.

Chapter 3. Planning for deployment

Host On-Demand provides access to host applications from a Web browser. The browser downloads the Host On-Demand Java applet from the Web server and then connects to any Telnet server to access host applications. The Host On-Demand applet needs configuration information to determine which host to connect to and other host session properties. This configuration information can be provided to the Host On-Demand applet from an HTML file or by using the Host On-Demand configuration server. The configuration server is a part of Host On-Demand that centrally stores session configuration information and user preferences by user and group IDs. Users then access session information and user preferences by contacting the configuration server. The configuration server is managed through the administration client. For information on configuring the Host On-Demand configuration server, see the online help.

You can create custom client HTML files using the Deployment Wizard. When creating these HTML files, you can choose from three different configuration models to specify how session configuration information and user preferences (for example, changes users make to session size and location, colors, etc.) are defined and managed: the HTML-based model, the configuration server-based model, and the combined model.

These models are described below. For detailed information on each model and benefits and limitations to using each model, see the online help.

Understanding the HTML-based model

If you choose the HTML-based model, all host session configuration information is contained in the HTML file itself, and nothing more is needed to define host sessions. Therefore, you are not required to use the configuration server to specify sessions, which means you do not have to open up a port on your firewall. If you allow users to save changes to the host session configuration information, their changes are stored on the local file system where the browser is running.

This option of defining configuration information in the HTML files is only available in clients that are created using the Deployment Wizard.

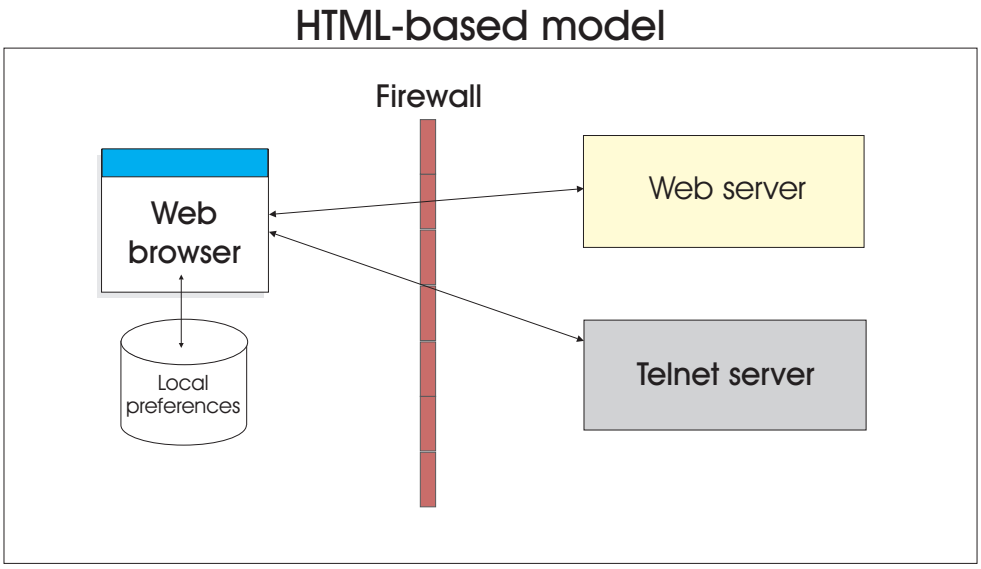

Figure 2. HTML-based model

Understanding the configuration server-based model

In the configuration server-based model, host session information is maintained on the configuration server using the Administration client, and the information is defined using a user and group structure. By default, the configuration server stores its data locally on the Host On-Demand server machine, though it can be configured to use LDAP instead. Users access their configurations using either custom HTML files created in the Deployment Wizard or by using one of several HTML files that are provided as part of Host On-Demand. User IDs are defined in the configuration server, and in most cases the user needs to log on to the Host On-Demand server before viewing his sessions. If administrators allow users to save changes, user preferences are stored in the configuration server by user ID. Because their customizations are saved on the configuration server, this model may be the best choice if users need to access their sessions from multiple machines.

By default, the Web browser communicates directly to the configuration server. If you communicate through a firewall, you need to open the configuration server's port on the firewall. You can also use the configuration servlet, through which the Web browser communicates with the configuration server. The connection from the Web browser to the configuration servlet is over HTTP or HTTPS, so the configuration server's port does not need to be opened on the firewall. See Configuring the configuration servlet for more information on using the configuration servlet.

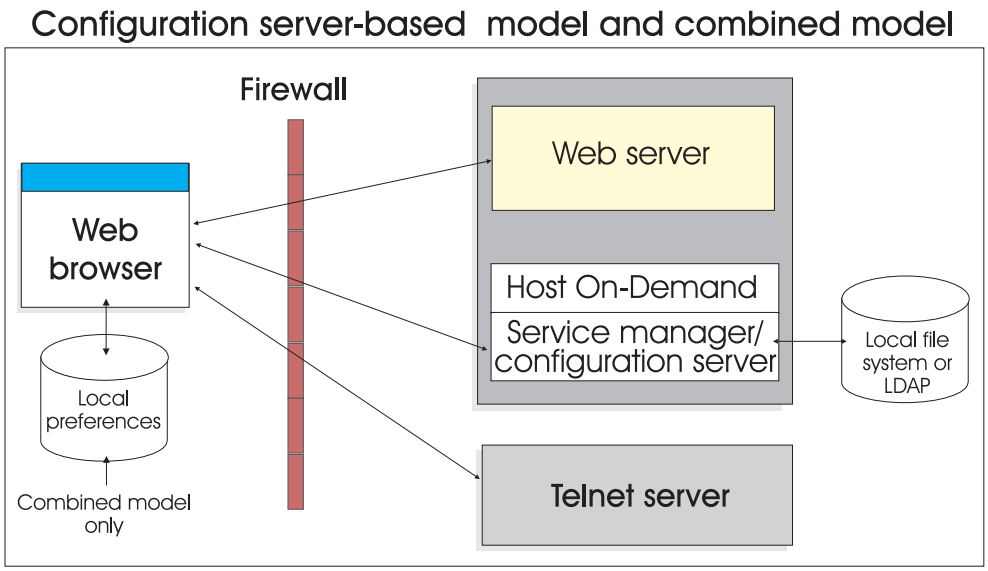

Figure 3. Configuration server-based model and combined model

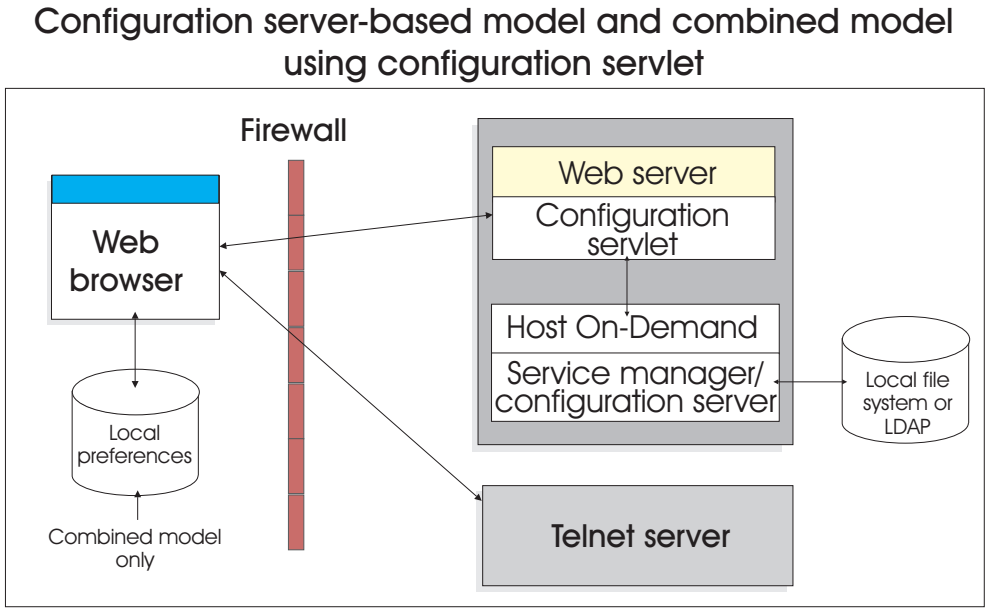

Figure 4. Configuration server-based model and combined model using configuration servlet

Understanding the combined model

Host On-Demand supports a combined model, where the host session information is defined in the configuration server (like the configuration server-based model) and user updates are saved on the user's machine (like the HTML-based model). In addition, like the HTML-based model, users of the combined model do not need to log on to the Host On-Demand server to view their sessions.

Client deployment considerations

Additionally, for client deployment considerations, you need to decide whether to use cached, download, or Web Start clients (see Chapter 12, “Using Host On-Demand emulator clients”, on page 95) and which version of Java to use (see Chapter 4, “Planning for Java 2 on the client”, on page 27).

Chapter 4. Planning for Java 2 on the client

There are several reasons why customers should consider making the transition from Java 1 browsers to Java 2-enabled browsers.

- Vendors who provide JVMs that use Java 1 are gradually withdrawing their support of these products. Withdrawing their support means no longer committing to fix bugs (including security bugs) or no longer making these products available.
- Java 2 is a proven technology and is actively supported by vendors.
- Java 2 provides capabilities that Java 1 lacks, including support for accessibility features.

Host On-Demand Version 8.0 continues to support both Java 1 and Java 2-enabled browsers. The Host On-Demand server has separate Java 1 and Java 2 versions of the Host On-Demand client. If a user is running a Java 1 browser and points to a Host On-Demand HTML page, then that user gets the Java 1 version of the Host On-Demand client. Likewise, if the user is running a Java 2-enabled browser and accesses a Host On-Demand HTML page, then that user (in most cases) gets the Java 2 version of the Host On-Demand client.

However, even though Host On-Demand has both a Java 1 and a Java 2 version of the client, the Java 2 version continues to acquire new features that the Java 1 version lacks, because the underlying Java 1 JVM does not contain the support required.

Host On-Demand is making the transition from Java 1 browsers to Java 2-enabled browsers easier in the following ways:

- The Deployment Wizard lets you configure an HTML file to indicate whether you want users to run it on a Java 1 browser only, on a Java 2-enabled browser only, or on either type of browser by creating the page as an Auto Detect page.
- Users running on the Windows platform can download the IBM Java 2 runtime for Windows directly from the Host On-Demand server. (This is the IBM 32-bit Runtime Environment for Java 2, v1.4).
- The online help for users has been expanded to provide more help for situations involving Java 1 and Java 2-enabled browsers.
- A parameter can be added to a Deployment Wizard generated HTML page that redirects the Java 1 browser running on Windows to the IBM Java 2 runtime.
- The Java 2 cached client now supports installation from a LAN or CD, sharing a cached client on Windows XP, and upgrading in the background. Also, the Java 2 download client can be run on a workstation on which the Java 2 cached client is installed.

This chapter provides detailed information related to running the Host On-Demand client on a Java 2–enabled browser.

- “Improvements to the cached client for Java 2” on page 28 describes improvements to the Host On-Demand Java 2 cached client.
- “Enhanced features provided by Java 2” on page 29 describes advanced features of the Host On-Demand client that are available only with a Java 2–enabled browser.

- "Upgrading your HTML files to support the Java 2 client" on page 77 discusses migrating HTML files from earlier versions of Host On-Demand to the current version.
- "Apple Mac OS X with Java 2" on page 29 discusses issues involved in using the Apple Mac OS X as a Host On-Demand client with Java 2.
- "Limitations with Java 2" on page 29 discusses limitations with using the Host On-Demand client with Java 2.
- "Java 1 and Java 2 versions of the Host On-Demand emulator client" on page 31 discusses the Java 1 and Java 2 versions of the Host On-Demand emulator client.
- "Browsers and Java 2 plug-ins" on page 31 discusses issues involved in using Java 1 browsers, Java 2–enabled browsers, and Java 2 plug-ins.
- "Host On-Demand Java level" on page 33 discusses issues surrounding the choice of a Host On-Demand Java level in the Deployment Wizard.
- "Obtaining a Java 2 plug-in for your clients" on page 33 discusses how to obtain the Java 2 plug-in.
- "Using the Java 2 plug-in" on page 33 describes how to perform various operations involving the Java 2 plug-in.

Improvements to the cached client for Java 2

The following improvements bring the Java 2 cached client up to the same level of user-friendliness and flexibility as the Java 1 cached client. With the Java 2 cached client, you can now do the following:

- Install the Java 2 cached client from a LAN drive or CD drive. For more information, refer to "Installing the cached client from a LAN or CD" on page 99.
- Share the Java 2 cached client between more than one user on Windows 2000 or Windows XP. For more information, refer to "Cached client support for Windows 2000 and Windows XP" on page 104.
- Remove the Java 2 cached client in one operation, without clearing the Java 2 plug-in's cache. For more information, refer to "Removing the cached client" on page 101.
- Upgrade the Java 2 cached client in the background.

  Note: The following restrictions apply:
  - Users upgrading the cached client from Host On-Demand 7 to Host On-Demand 8 cannot choose to upgrade in the background.
  - A few Java 2 cached client types cannot be upgraded in the background. See "Improvement support limitations".

Almost all Host On-Demand Java 2 cached clients support these improvements. The Java Web Start client also supports these improvements.

Improvement support limitations

The following types of Java 2 cached clients do not support the improvements to the Java 2 cached client:

- Java 2 Administration cached clients
- Java 2 cached clients on the Apple Mac OS X
- Java 2 emulator cached clients that have the JavaScript Session Manager API enabled

Enhanced features provided by Java 2

Using a Java 2–enabled browser with a Java 2 plug-in, you can take advantage of the following advanced features offered by the Host On-Demand client. For more information on Java 2-enabled browsers, refer to "Browsers and Java 2 plug-ins" on page 31.

- Web Start client
- Support for the Secure Shell (SSH) for VT Display sessions and secure File Transfer Protocol (sftp) sessions
- Auto IME/On-the-Spot Conversion
- Print Screen Enhancements
- Internet Protocol Version 6 (IPv6)
- Accessibility features (requires Java version 1.4 or later)
- For bidirectional languages, support is now provided for OS/400 Coded Character Set Identifiers (CCSIDs) for displaying Unicode characters.

Apple Mac OS X with Java 2

Host On-Demand Mac OS X emulator and database clients support Safari and the Mac version of Internet Explorer. Host On-Demand does not support the administration clients on Mac OS X. If your users use Safari, they should upgrade to JRE 1.4.1, available at http://www.apple.com.

Mac OS X does not support Java 1 browsers.

Limitations with Java 2

This section discusses a number of client limitations to be aware of with Java 2.

Downloading a client with Java 2

The following sections discuss the limitations in downloading a client with Java 2.

Cannot download a component not in the preload list

With the Java 2 download client, a user cannot download a Host On-Demand client component that is not in the original preload list. Consequently, you must specify all the components that your users might require in the preload list.

This limitation is caused by a conflict between the method used by a download client to download components not on the preload list and security restrictions imposed by the Java 2 plug-in.

HTML files do not contain some components

With Java 2, the default download client HTML files (HOD_*xx*.html, where *xx* is the two-letter language suffix) do not contain the following client components:

- 5250 file transfer
- Import/export
- SLP
- Thai sessions
- FTP Codepage Converter
- Bidirectional sessions
- 5250 Hindi sessions
- DBCS sessions using user-defined character settings

IBM removed these less frequently used components from the preload list of the Java 2 default download HTML files to shorten download time. However, with the Java 2 download client, any component not in the preload list cannot be downloaded later.

If you want some or all of these components to be in the preload list, perform one of the following actions:

- Use the Deployment Wizard to create a download client Java 2 HTML file that contains exactly the components that you need.
- Use the default HTML file for the cached client (HODCached_xx.html, where xx is the two-letter language suffix) instead of the default HTML file for the download client.
- Use the debug version of the default download client (HODDebug_en.html, and so on). The debug version contains all the components. However, the debug version of the default download client is larger than the non-debug version.

Mac OS X limitations

Mac OS X does not support the Java 2 cached client improvements described in "Improvements to the cached client for Java 2" on page 28. For more information, refer to "Cached client support for Mac OS X (Java 2 clients only)" on page 105.

Slightly slower startup times with Java 2 clients

With a Java 2–enabled browser, the Host On-Demand client starts a little more slowly (5 to 15 seconds slower, depending on the workstation type) than with a Java 1 browser. The delay is caused by the system loading the Java 2 plug-in.

With a Java 2–enabled browser, a host session on the Host On-Demand client desktop can take a little longer in starting (a few seconds slower) than with a Java 1 browser.

Limitations of specific Java 2 plug-ins

The Sun Java 2 plug-in has a limitation with Hindi character conversion. If you need Hindi character conversion, use the IBM Java 2 plug-in.

Limitations with customer-supplied applets and Java 2

If a user runs a customer-supplied applet (that is, an applet written by your company or a third party) with a session (such as 3270 Display) launched from a Java 2 Host On-Demand client, and if this applet requires any Java 2 permissions, then you must take one of the following actions to meet the security requirements of Java 2:

- The applet must be archived in a signed Java 2 .JAR file.
- The permissions must previously have been granted on the workstation using the Java 2 Policy Tool that is provided with the Java 2 plug-in.

If you do not meet the security requirements of Java 2, the applet silently fails.

Limitations with restricted users and Java 2

Restricted users do not have the authority to install the Java 2 plug-in. A user with administrative authority must install the Java 2 plug-in.

Java 1 and Java 2 versions of the Host On-Demand emulator client

The Host On-Demand server has in its publish directory two versions of the Host On-Demand emulator client: a Java 1 version and a Java 2 version. The Java 1 version consists of a complete set of Host On-Demand client components compiled with a Java 1 compiler. The Java 2 version consists of a complete set of Host On-Demand client components compiled with a Java 2 compiler.

See "Terms relating to Java 1 and Java 2" on page x for specifics on the terminology used throughout this document.

When one of your users starts an emulator download client or installs an emulator cached client, Host On-Demand determines which version (Java 1 or Java 2) of the client to start or install. The two most important factors in this determination are:

- The type of browser that the user is running (Java 1 or Java 2-enabled)
- The Host On-Demand Java level of the Host On-Demand HTML file (Java 1, Java 2, or Auto Detect)

For more information on Host On-Demand Java levels and on how Host On-Demand determines which version of the emulator client to run, refer to "Host On-Demand Java level" on page 33.

For more information on Java 1 and Java 2-enabled browsers, refer to the next section, "Browsers and Java 2 plug-ins".

For more information on how to determine which version (Java 1 or Java 2) of the emulator client is running, refer to Using the Java 2 plug-in in the online help.

Browsers and Java 2 plug-ins

This section discusses issues involved in using Java 1 browsers, Java 2–enabled browsers, and Java 2 plug-ins.

Note: IBM recommends that you install only one Java 2 plug-in on a Host On-Demand client workstation. Java 2 plug-ins from different vendors, or even different versions of the same vendor's Java 2 plug-in, do not always work well together on the same workstation.

If you already have multiple Java 2 plug-ins installed on a client, refer to Using the Java 2 plug-in in the online help for instructions on how to proceed.

Java 1 and Java 2-enabled browsers

A Java 1 browser typically has a Java 1 JVM included with the browser. The Java 1 JVM is capable of running classes compiled using Java 1 (for example, Java 1 applets) but it is not capable of running classes compiled using Java 2. Examples of Java 1 browsers are Microsoft Internet Explorer without the Java 2 plug-in installed and Netscape 4.7.

In contrast, a Java 2–enabled browser does not have a JVM included with it. It can display HTML files on its own, but it needs a separate Java 2 plug-in installed to launch a Java applet such as the Host On-Demand client. This Java 2 JVM is capable of running either Java 1 or Java 2 applets, but it is better at running Java 2 applets. Examples of Java 2–enabled browsers are Netscape Version 6.0, Netscape Version 7.0, and Microsoft Internet Explorer with the Java 2 plug-in installed.

Browsers and plug-ins supported by Host On-Demand clients

For a list of browsers and Java 2 plug-ins supported by Host On-Demand clients, refer to "Supported browsers and Java 2 plug-ins" on page 20.

Users with client workstations running Windows can download the IBM Java 2 plug-in v1.4 from any Host On-Demand server. See "Obtaining a Java 2 plug-in for your clients" on page 33.

As vendors of Java 2 plug-ins such as Sun, IBM, and Hewlett-Packard publish new versions of their Java 2 plug-ins, and as IBM extends Host On-Demand to support these new versions, IBM will announce support of the new versions on the Host On-Demand Web site at:
http://www.ibm.com/software/webservers/hostondemand.

Microsoft Internet Explorer with a Java 2 plug-in

When a Java 2 plug-in is properly installed and configured on a Windows client workstation, Microsoft Internet Explorer can function as either a Java 1 browser or as a Java 2-enabled browser, depending on how Host On-Demand chooses to launch the client.

Internet Explorer's default JVM must be the Java 1 JVM

When a Java 2 plug-in is installed on Windows to use with Microsoft Internet Explorer, the Host On-Demand client expects *both* of the the following settings to be configured:

- The Java 2 plug-in should be configured so that it is *not* the default Java Runtime for Internet Explorer.
- Internet Explorer should be configured so that it does *not* use the Java 2 plug-in for the <applet> HTML tag.

For instructions on how to check and change these settings see Setting the default Java Runtime for a Java 2-enabled browser in the online help.

Not all Java plug-ins have the first setting listed above, and not all versions of Internet Explorer have the second setting. If the plug-in or the version of Internet Explorer that you are using does not provide a way to configure this setting, then the default configuration of that tool is probably the correct one.

Why Internet Explorer's default JVM must be the Java 1 JVM: The default JVM (also called the default Java Runtime) is a setting in Internet browsers that identifies the JVM that the browser uses when an applet is launched on the browser in the default manner (using the <applet> HTML tag).

Internet Explorer initially has its default JVM set to the Java 1 JVM that is included with Internet Explorer. But after the Java 2 plug-in is installed there might be (depending on the version of the plug-in) an option that allows you to set the default JVM of Internet Explorer to be the Java 2 plug-in. In addition, Internet Explorer itself might have (depending on its version) a similar setting that allows you to set the default JVM to the Java 2 plug-in.

IBM strongly recommends that you verify that the default JVM of Internet Explorer is set to the Java 1 JVM. You should check both the plug-in's configuration and the browser's configuration.

When a Java 2 plug-in is installed and the default JVM is the Java 1 JVM, then the Host On-Demand client is able to use Internet Explorer flexibly, either as a Java 1

browser or as a Java 2-enabled browser. In contrast, when Internet Explorer's default JVM is set to be the Java 2 plug-in, then the Host On-Demand client might not be able to run the HTML file correctly.

Netscape Versions 6 and 7 with a Java 2 plug-in

To run a Java applet on Netscape Version 6 or 7, you must install a Java 2 plug-in. Netscape Versions 6 and 7 do not include and cannot use a Java 1 JVM.

Consequently, Host On-Demand expects you to configure the Java 2 plug-in so that it *is* the default Java Runtime for Netscape. For instructions on how to check or change this setting, refer to the Setting the default Java Runtime for a Java 2-enabled browser topic in the online help.

Unlike Internet Explorer, the Netscape Version 6 or 7 browser itself does not have a setting for changing the default JVM. You need only to verify that the Java 2 plug-in's setting is correct.

Not all Java plug-ins have this setting. If the plug-in does not provide a way to change this setting, then the default configuration is correct.

Host On-Demand Java level

Host On-Demand Java level identifies the type of browser that a client should use to run the generated Host On-Demand HTML file. The three choices are Java 1, Java 2, and Auto Detect. For explanations of these three options, refer to "Host On-Demand Java level" on page 85.

Obtaining a Java 2 plug-in for your clients

On all supported platforms, the Host On-Demand server includes a downloadable install image of the IBM Java 2 plug-in for the Microsoft Windows platform. The plug-in is called the IBM 32–bit Runtime Environment for Java 2.

Consequently, any client running on a supported Windows platform can attach to a Host On-Demand server, download the install image, and install the IBM Java 2 JRE. For instructions see Downloading and installing the IBM Java 2 plug-in for the Microsoft Windows platform in the online help.

Note: Restricted users, such as restricted users sharing a cached client on Windows 2000 or Windows XP, or restricted users on a Linux or AIX workstation, cannot install the Java 2 plug-in. See "Limitations with restricted users and Java 2" on page 30. The Java 2 plug-in must be installed by a user with administrator authority on the workstation.

For the Sun Java 2 plug-ins, see the Sun Microsystems Web site at http://www.javasoft.com.

Using the Java 2 plug-in

Using the Java Plug-in Control Panel

The Java Plug-in Control Panel is launched differently depending on the client platform and on the vendor of the plug-in. For more information, refer to Launching the Java 2 Plug-in Control Panel in the online help.

Chapter 5. Planning for security

Whether you are implementing Host On-Demand purely within your corporate network, or you are using it to provide access to your host systems over the Internet, security is a concern. This chapter provides an overview of Host On-Demand security.

- Transport Layer Security (TLS) and Secure Sockets Layer (SSL) security. Provides encryption, certificate-based authentication, and security negotiations over an established Telnet or FTP connection. See "TLS and SSL for Host On-Demand" on page 37 for details.
- Secure Shell (SSH). Provides secure sessions over a non-secure network. Includes secure remote login, strong authentication of server and client, several user authentication methods, encrypted terminal sessions, and secure file transfers. See "Secure Shell (SSH)" on page 41.
- Should I use SSH, or TLS and SSL? Comparison of these security protocols. See "Should I use SSH, or TLS and SSL?" on page 46.
- The Redirector. Supports TLS and SSL between Host On-Demand clients and the Host On-Demand server. See "The Redirector" on page 46 for details.
- Firewalls. You can configure Host On-Demand to go through a firewall. See "Using Host On-Demand with a firewall" on page 47 for details.
- User ID security. Includes Web Express Logon, Native Authentication, and Windows Domain logon. See "User ID security" on page 52 for details.

Transport Layer Security (TLS) and Secure Sockets Layer (SSL) security

How TLS and SSL security work

The TLS and SSL security protocols are very similar; in fact, TLS is based on the SSL protocol. TLS differs from SSL mainly in the initial handshake protocol for establishing client/server authentication and encryption. It is also more extensible than SSL. Although they cannot interoperate, TLS provides a mechanism by which a TLS 1.0 implementation can revert to SSL 3.0. For detailed information on TLS, see the description of *The TLS Protocol Version 1.0* at http://www.ietf.org/rfc/rfc2246.txt.

The TLS protocol uses public-key and symmetric-key cryptographic technology. Public-key cryptography uses a pair of keys: a public key and a private key. Information encrypted with one key can be decrypted only with the other key. For example, information encrypted with the public key can be decrypted only with the private key. Each server's public key is published, and the private key is kept secret. To send a secure message to the server, the client encrypts the message by using the server's public key. When the server receives the message, it decrypts the message with its private key.

Symmetric-key cryptography uses the same key to encrypt and decrypt messages. The client randomly generates a symmetric key to be used for encrypting all session data. The key is then encrypted with the server's public key and sent to the server.

TLS provides three basic security services:

Message privacy
Achieved through a combination of public-key and symmetric-key encryption. All traffic between a client and a server is encrypted using a key and an encryption algorithm negotiated during session setup.

Message integrity
Ensures that session traffic does not change en route to its final destination. TLS and SSL use a combination of public/private keys and hash functions to ensure message integrity.

Mutual authentication
Exchange of identification through public-key certificates. The client and server identities are encoded in public-key certificates, which contain the following components:

- Subject's distinguished name
- Issuer's distinguished name
- Subject's public key
- Issuer's signature
- Validity period
- Serial number

You can also use secure HTTP (HTTPS) to ensure that a client's security information is not compromised as it is downloaded from a server.

Certificates

Security is controlled by digital certificates that act as electronic ID cards. The purpose of a certificate is to assure a program or a user that it is safe to allow the proposed connection and, if encryption is involved, to provide the necessary encryption/decryption keys. They are usually issued by Certificate Authorities (CAs), which are organizations that are trusted by the industry as a whole and whose business is the issuing of Internet certificates. A CA's certificate, which is also known as a root certificate, includes (among other things) the CA's signature and a validity period.

Encryption and authentication are performed by means of a pair of keys, one public, one private. The public key is embedded into a certificate, known as a site or server certificate. The certificate contains several items of information, including the name of the Certificate Authority (CA) that issued the certificate, the name and public key of the server or client, the CA's signature, and the date and serial number of the certificate. The private key is created when you create a self-signed certificate or a CA certificate request and is used to decrypt messages from clients.

A TLS or SSL session is established in the following sequence:

1. The client and the server exchange hello messages to negotiate the encryption algorithm and hashing function (for message integrity) to be used for the session.
2. The client requests an X.509 certificate from the server to prove its identity. Optionally, the server can request a certificate from the client. Certificates are verified by checking the certificate format and the validity dates and by verifying that the certificate includes the signature of a trusted certificate authority (or is self-signed).
3. The client randomly generates a set of keys that is used for encryption. The keys are encrypted with the server's public key and securely communicated to the server.

TLS and SSL for Host On-Demand

There are three areas where you can configure security for Host On-Demand: session security, Web server security, and configuration security.

Session security

Host On-Demand can use two protocols to provide security for emulator and FTP sessions.

- The TLS protocol provides communications privacy across a TCP/IP network. TLS is designed to prevent eavesdropping, message tampering, or message forgery. TLS also provides a framework that allows new cryptographic algorithms to be incorporated easily. Host On-Demand supports encryption of emulation and FTP sessions and server/client authentication according to *TLS Protocol Version 1.0* standard (available at http://www.ietf.org/rfc/rfc2246.txt).
- The SSL protocol provides encryption and authentication on connections across a TCP/IP network, using X.509 certificates. Host On-Demand supports encryption of emulation and FTP sessions and server/client authentication according to the SSL Version 3.0 standard.

Support is provided for the following:

- RSA type-4 data encryption on connections between the Host On-Demand sessions and Telnet or FTP servers that support TLS version 1.0 and SSL version 3
- X.509 certificates
- Bulk encryption algorithms using keys up to 168 bits in length
- Authentication algorithms using keys up to 1024 bits in length
- Server and client authentication
- Support for storage and use of client certificates on the client system
- Optional prompting of user for client certificate when requested by server

For Host On-Demand, you can use a CA's certificate, but you can also create your own self-signed certificate, as described in the Using a self-signed certificate topic in the online help.

A graphical Certificate Management utility (available on Windows and AIX platforms) is provided to:

- Create certificate requests
- Receive and store certificates
- Create self-signed certificates

IKEYCMD is a tool, in addition to the Certificate Management utility, that you can use to manage keys, certificates, and certificate requests. IKEYCMD is functionally similar to Certificate Management and is meant to run from the command line without a graphical interface. For more information, refer to Appendix B, "Using the IKEYCMD command-line interface", on page 167.

To support TLS and SSL services, Host On-Demand uses three databases:

HODServerKeyDb.kdb

You create the HODServerKeyDb.kdb the first time you configure TLS or SSL for the Host On-Demand Redirector. This database contains the server's private key and certificate as well as a list of CA (or signer) certificates. These CAs are considered *well-known* and are *trusted* by the Host On-Demand server. You can add certificates from other CAs

(unknown CAs) and certificates that you create and sign yourself (self-signed) to this database. Refer to “The Redirector” on page 46 for more information.

CustomizedCAs.p12

The CustomizedCAs.p12 is a PKCS#12 format file that contains the root certificates of unknown CAs and self-signed certificates that are not in the WellKnownTrusted list. If you use a self-signed certificate or a certificate from an unknown authority (CA), you must create or update the CustomizedCAs.p12. Host On-Demand does not install a CustomizedCAs.p12 file by default.

The CustomizedCAs.p12 file is a newer version of the CustomizedCAs.class file, which you may have created with an earlier release of Host On-Demand. The CustomizedCAs.class file supports Host On-Demand Version 7 and earlier clients, and is located in your publish directory by default. If you are running Windows or AIX, when you upgrade to version 8, the Host On-Demand installation automatically detects the CustomizedCAs.class file, creates the new CustomizedCAs.p12 file, and places it in the publish directory. Both files remain in your publish directory and are available to clients of different versions. If you have an separate user publish directory and not the default publish directory, the Host On-Demand installation will not be able to detect the CustomizedCAs.class file and you will need to run the migration tool manually on the command line. Refer to “Migrating from CustomizedCAs.class to CustomizedCAs.p12” on page 74 in ″Upgrading from earlier versions of Host On-Demand″ for more information.

If you create the CustomizedCAs.p12 file for the first time using the Host On-Demand Version 8 Certificate Management utility (IKEYMAN), you will also want to have the older CustomizedCAs.class file in your publish directory so that older clients can still operate with the new server. Also, when you subsequently update the CustomizedCAs.p12 file, you will want to make sure these changes are picked up by the CustomizedCAs.class file. For Windows platforms, if these files are in the default publish directory, c:\Program Files\IBM\HostOnDemand\HOD, each time you open IKEYMAN to update the CustomizedCAs.p12 file and then close IKEYMAN, the CustomizedCAs.class file is automatically updated along with the CustomizedCAs.p12 file. If these files are not in the default publish directory, you need to manually run the reverse-migration tool from your publish directory using the following command. The command appears on two lines, but you should type it on one line.

```
..\jre\bin\java -cp ..\lib\sm.zip com.ibm.eNetwork.HOD.convert.CVT2SSLIGHT
CustomizedCAs.p12 hod CustomizedCAs.class
```

On AIX, for the CustomizedCAs.class file to pick up the changes you make to the CustomizedCAs.p12 file, you must run this reverse-migration tool manually from your publish directory using the following command:

```
You should type the following on one line:
../jre/bin/java -cp ../lib/sm.zip com.ibm.eNetwork.HOD.convert.CVT2SSLIGHT
CustomizedCAs.p12 hod CustomizedCAs.class
```

CustomizedCAs.class

The CustomizedCAs.class is a Java class file that contains the certificates of unknown CAs and self-signed certificates that are not in the WellKnownTrusted list. If you use a self-signed certificate or a certificate from an unknown authority (CA), you must update the CustomizedCAs.class file. However, note that you can no longer create or

update the CustomizedCAs.class file using the Certificate Management utility on Windows or AIX platforms. In Host On-Demand Version 8, you can only create a newer version of this file called CustomizedCAs.p12. All clients still support the older format, however. For more information, refer to the description of CustomizedCAs.p12 above.

WellKnownTrustedCAs.class
The WellKnownTrustedCAs.class is a Java class file supplied by Host On-Demand that contains the public certificates of all the CAs that Host On-Demand trusts. You should not modify this file.

WellKnownTrustedCAs.class and CustomizedCAs.p12 and/or CustomizedCAs.class must be present in the Host On-Demand publish directory. The Host On-Demand client uses these files to trust the server's certificate during the TLS or SSL handshake.

Basic TLS or SSL enablement for Host On-Demand clients

When you select the TLS or SSL protocol for the Host On-Demand client, a basic TLS or SSL session is established. During the TLS or SSL negotiation process, the server presents its certificate to the client. With basic TLS or SSL enablement, the certificate must be signed by an authority that the client trusts. The client checks WellKnownTrustedCAs.class first, followed by the CustomizedCAs.p12 or the CustomizedCAs.class. The client rejects the session if it does not find the signer in these files. If the client finds the signer in these files, the session is established. This is basic Server Authentication. Host On-Demand allows you to configure a more enhanced form of Server Authentication in its client configuration. Refer to the following section for more information.

Server authentication
Encrypting the data exchange between the client and the server does not guarantee the client is communicating with the correct server. To help avoid this danger, you can enable server authentication, so that the client, after making sure that the server's certificate can be trusted, checks whether the Internet name in the certificate matches the Internet name of the server. If they match, the TLS or SSL negotiation will continue. If not, the connection ends immediately. See server authentication in the online help for more information.

Client authentication
Client authentication is similar to server authentication except that the Telnet server requests a certificate from the client to verify that the client is who it claims to be. Not all servers support client authentication, including the Host On-Demand Redirector. To configure client authentication, you must do the following:

- obtain certificates for clients
- send the certificates to the clients
- configure the clients to use client authentication

Refer to configuring clients to use client authentication in the online help for more information.

Express Logon
There are two types of Express Logon:

- Web Express Logon: Web Express Logon allows users to log on to host systems and host applications without having to provide a user ID and password. This feature works in conjunction with your network security application by acquiring the user's network credentials and mapping them to their host credentials, eliminating the need to log on multiple

times. Depending on your host, the logon automation process can be macro-based or connection-based. For more information, refer to the `Web Express Logon Reference`.

- Certificate Express Logon: Certificate Express Logon is macro-based and also allows users to log on without having to enter a user ID and password. It is functionally similar to Web Express Logon, although it requires you to configure your session for TLS or SSL and client authentication, and the Communications Server must support and be configured for Express Logon. For more information, refer to Express logon in the online help.

TLS-based Telnet security

Telnet-negotiated security allows the security negotiations between the client and the Telnet server to be done on the established Telnet connection. You can configure Telnet-negotiated security for Host On-Demand 3270 display and printer sessions.

The Telnet server must support TLS-based Telnet security (as described in the IETF Internet-Draft *TLS-based Telnet Security*, available at http://www.watersprings.org/pub/id/draft-ietf-tn3270e-telnet-tls-06.txt) for the Host On-Demand clients to use Telnet-negotiated security. The Communications Server for OS/390 Version 2 Release 10 and later supports TLS-based Telnet security. Communications Server for OS/390 documentation refers to Telnet-negotiated security as "negotiable SSL."

For more information regarding Telnet-negotiated security, see the Telnet-negotiated security overview in the online help. Refer to your Telnet server's documentation for more information about configuring TLS or SSL on the Telnet server, and refer to the Security topic in the online help for more information about configuring a client to connect to a secure Telnet server.

Examples of when to use session security

Refer to the following examples as situations where you might want to use session security:

- Allowing customers to order your products over the Internet. In this situation, you want to make sure the information customers give you, such as a credit-card number, is encrypted so that it cannot be stolen. You also want to make sure information you give to customers is protected.
- Giving your suppliers or business partners access to information on your host computers. You do not want anyone else to be able to access this data.
- Allowing your staff to have access to your host-computer information from remote sites or when they are traveling.
- Giving doctors access to patient records from wherever they are and making sure that unauthorized people cannot access these records.

Web server security

You can configure your Web server to use TLS or SSL (HTTPS), so that the data stream from your Web server to your browser is encrypted. See your Web server documentation for more information about configuring your Web server for TLS or SSL. Once the client is loaded in a browser, however, it communicates directly with the host. You can configure Host On-Demand to provide TLS or SSL security to your host sessions. For more information, see Configuring TLS and SSL in the online help.

Configuration security

If you use the HTML model, your session configuration information will be encrypted if you use HTTPS. For all other models, you need to configure Host On-Demand to use the configuration servlet over HTTPS (after configuring your Web application server) to encrypt the session configuration instead of communicating directly with the configuration server. See "Installing the configuration servlet" on page 69 in this guide for more information about installing the configuration servlet, and see configuring the configuration servlet in the online help for more information about configuring clients to use the configuration servlet.

Secure Shell (SSH)

What is the Secure Shell (SSH)?

The Secure Shell (SSH) is a set of protocols for implementing secure sessions over a non-secure network (such as a standard TCP/IP network). In order to use SSH, you must set up SSH server software on the host. Security features include the following:

- Secure remote login
- Strong authentication of server and client
- Several user authentication methods
- Encrypted terminal sessions
- Secure file transfers

SSH: Level and features supported by Host On-Demand

Host On-Demand supports SSH as an option on the following session types:

- VT Display sessions
- File Transfer (sftp) sessions

The implementation of SSH in Host On-Demand is a subset of SSH Version 2. Host On-Demand does not support earlier versions of SSH, such as Version 1.3 or Version 1.5. The following table summarizes this information:

Table 10.

Version of SSH	Supported by Host On-Demand
SSH Version 2.0	Yes (subset)
SSH Version 1.5	no
SSH Version 1.3	no

The following subsections describe for each protocol in SSH Version 2.0 the features that Host On-Demand supports, or the features that Host On-Demand does not support.

SSH Transport Protocol

For the SSH Transport Protocol, Host On-Demand supports the following algorithms. The same algorithms are supported for sending files (client to server) and receiving files (server to client).

Table 11.

Category	Algorithm supported
Compression:	none
Encryption:	3des-cbc
Data Integrity:	hmac-sha1
Key Exchange:	diffie-hellman-group1–sha1
Public Key:	ssh-dss (same as DSA)

SSH Authentication protocol

For the SSH Authentication protocol Host On-Demand supports the following authentication methods:

- Public key
- Password

SSH Connection protocol

Host On-Demand does not support the following features in the SSH Connection protocol:

- X11 forwarding
- Environment Variable Passing
- Remote Command Execution
- Windows Dimension Change Message
- Signals
- TCP/IP Port Forwarding

SSH File Transfer protocol

For the SSH File Transfer Host On-Demand supports only the following transfer modes:

- Binary

  That is, Host On-Demand does not support character-mode file transfers for SSH.

Host On-Demand supports only binary mode transfer for SSH.

Host On-Demand does not support the following features in the SSH File Transfer protocol:

- Encoding of filenames in UTF-8 format
- Newline extension
- Operations that use symbolic links

Host On-Demand client requirements for SSH support

For SSH support Host On-Demand requires the following configuration on the client workstation:

- A Java 2–enabled browser
- The Java Cryptography Extension (JCE)

SSH is not supported with a Java 1 browser because Java 1 does not support the JCE.

The JCE is included as part of the IBM 32-bit Runtime Environment (JRE) for Java 2, v1.4 (for Windows platforms). This version of the Java 2 JRE is included with Host On-Demand and can be downloaded by Windows platform clients from the Host On-Demand server. For more information, refer to "Obtaining a Java 2 plug-in for your clients" on page 33.

The JCE is also included in Sun Java 1.4.

If you use Java 1.3 then you have have to first install Java 1.3 and then install the JCE. You cannot use Java 1.2.

Authentication for SSH

This section describes Host On-Demand's support of public-key authentication and password authentication on the client.

Both types of authentication can be configured at once

Host On-Demand allows both public-key authentication and password authentication to be configured on the client at the same time. At run time:

- If public-key authentication is configured, then Host On-Demand tries this type of authentication with the host first. If public-key authentication is not configured or if it is configured and fails, then Host On-Demand moves on to password authentication.
- For password authentication, Host On-Demand looks for a password in the session configuration. If no password is found, Host On-Demand prompts the user for a password. Once a password is received, Host On-Demand then tries password authentication with the host.
- If password authentication fails, then Host On-Demand displays an error message.

Public-key authentication

Configuring public-key authentication on the server: The server configuration for public-key authentication differs depending on the vendor or source of the SSH support. Refer to the documentation for your SSH server software for information on how to configure the SSH server for the public-key authentication method.

Generating a public-key file on the client and transferring it to the server: Public key authentication for SSH requires that the server knows the public key of the client. Here is an overview of the method for generating this public key and making it available to the server with Host On-Demand. A detailed explanation of each step follows this overview:

1. Run the Java Cryptographic Extension (JCE) keytool utility to generate a keystore containing the client's public key.
2. Place the file in the proper subdirectory on the client workstation.
3. Configure the Host On-Demand session configuration parameters for SSH.
4. Run the Host On-Demand Export Public Key utility to export the public key to a plain-text file.
5. Transfer the plain text file to the host.

The first step is to use the keytool utility in the JCE to generate a keystore containing a pair of keys for the client (a public key and a private key). To generate the keystore, invoke the keytool utility as follows:

```
keytool -genkey
```

For example, on a Windows platform you might type the following:

```
c:\program files\ibm\java14\jre\bin\keytool.exe -genkey
```

The keytool utility then prompts you for the following information:

- A password for the keystore
- Information routinely requested for public-private key pairs, including:
  - User's first and last name
  - Organizational unit
  - Organization
  - City or Locality
  - State or Province
  - Two-letter country code
- A password for the public-private key pair, which might be the same as the password for the keystore

When invoked with only the -genkey option, as above, the keytool utility generates the items listed below. These are the default values generated by the keytool utility and are also the default values expected by Host On-Demand configuration.

- A keystore with the name `.keystore`.

  By default, the keytool utility generates this file in the directory named in the Java system property `user.home`. For example, for the Windows platform, the file would be generated in the following directory:

  ```
  c:\Documents and Settings\username
  ```

  where *username* is the user name.
- In the keystore, a 1024-bit DSA key pair (a public key and associated private key) with the key alias `mykey`. Host On-Demand supports 1024-bit DSA keys only.

To generate a keystore with a non-default filename, key alias, store password, and alias password, invoke the keytool utility with the following command. Note that the command appears in this document on two lines; however, you should type it all on one line.

```
keytool -genkey -keystore MyKeystoreFile -alias MyAlias
        -storepass MyKeystorePassword -keypass MyKeyPassword
```

Run the keytool utility with no options specified to see all the possible options.

The second step is to place the keystore file in the proper subdirectory on the client workstation. As mentioned above, the default file name is `.keystore` and the default subdirectory is the path stored in the `user.home` Java system property. In any case, you should use the same file name and path that you plan to specify in the session configuration.

The third step is to configure the Host On-Demand session parameters for SSH. As mentioned above two Host On-Demand session types support SSH:

- VT Display
- File Transfer (sftp)

You will need to specify the following information (or you can accept the default values):

- Path and file name for the keystore.

  - The default is the file `.keystore` in the directory pointed to by the Java system property `user.home`.
- Password for the keystore.
  - If no password is specified in the configuration then when the session is started the Host On-Demand client will display a popup window prompting for the password.
- Key alias.
  - The default is `mykey`.
- Password for the key alias.
  - If no password is specified then when the session is started Host On-Demand will attempt to read the public key information using a null password (no password).
  - If the attempt to read the public key information using a null password fails then the Host On-Demand client will attempt to read the public key information using the same password as the password for the keystore.
  - If the attempt to read the public key information using the KeyStore Password fails then Host On-Demand client will prompt the user for the password.

For more information, refer to SSH configuration in the online help.

The fourth step is to run the Host On-Demand Export Public Key utility in order to export the public key to a plain-text file. This utility is not a stand-alone utility but rather is integrated with the session configuration. To run the utility, go to the SSH configuration panel in the session configuration, the same panel where you specified the path and file name for the keystore, and click Export Public Key. Follow the instructions to export the public key to a plain text file.

The fifth step is to transfer the plain text file to the host. You should use a secure method for transferring the plain text file to the host, such as one of the following:

- SSH file transfer (sftp)
- Diskette

Configuring public-key authentication on the client: To configure the client for public-key authentication, a keystore containing the client's public and private key information must be placed either:

- On the client
- On a drive reachable by the client, such as a network drive.

Password authentication

Configuring password authentication on the server: The server configuration for password authentication differs depending on the vendor or source of the SSH support. Refer to the documentation for your SSH server software for information on how to configure the SSH server for the password authentication method.

Configuring password authentication on the client: You do not need to configure the client for password authentication. The Host On-Demand client will look for the password in the session configuration information. If no password is found, then Host On-Demand will prompt the user for a password.

Should I use SSH, or TLS and SSL?

Both SSH and TLS/SSL provide secure sessions. Which protocol is better for you depends on the characteristics of the system that you support:

- SSH is easier to set up, because it does not require certificates on the client or the host.
- SSH requires the presence of an SSH server on the host.
- Host On-Demand Version 8 supports SSH only on VT and sftp sessions; that is, there is no SSH support on 3270 and 5250 sessions.

The Redirector

Why use the Redirector?

If your Telnet server does not support TLS or SSL, and you are running Host On-Demand on Windows NT, Windows 2000, or AIX on Netscape Communicator 4 or Internet Explorer 4 or later browsers, you can configure the Host On-Demand Redirector to provide TLS or SSL support. The Redirector, which resides on the Host On-Demand Server, provides support for TLS and SSL security between clients and the Host On-Demand server.

Many Telnet servers support TLS or SSL (for example, IBM Communications Servers on zSeries, iSeries, AIX, NT, and OS/2). If your Telnet server supports TLS or SSL, we strongly recommend using your Telnet server. If your Telnet server does not support TLS or SSL, the Communications Server for AIX Redirector offers a more scalable alternative to the Host On-Demand Redirector.

The Redirector acts as a transparent Telnet proxy that uses port remapping to connect the Host On-Demand server to other Telnet servers. Each defined server can configure a set of local-port numbers. Instead of connecting directly to the target Telnet server, a client connects to the Host On-Demand server and port number. The Redirector maps the local-port number to the host-port number of the target and makes a connection.

The recommended solution for a Telnet proxy is to use Load Balancer, a feature of WebSphere Application Server's Edge Components, or a similar product that provides address translation as part of the overall firewall solution, instead of the Host On-Demand Redirector.

Redirector load capacity

For Redirector load capacity recommendations, refer to the Readme.

How the Redirector works

The following scenario shows how the Redirector works.

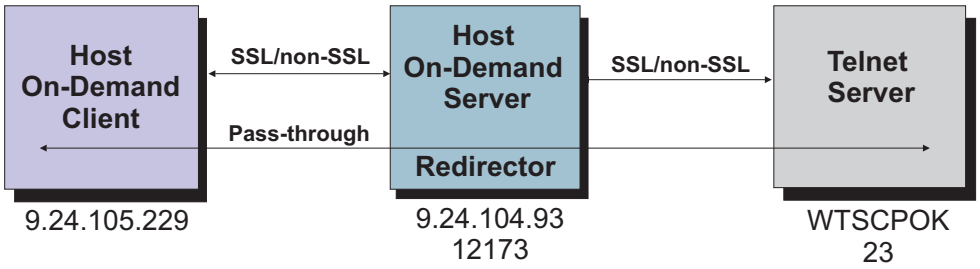

Figure 5. How the Redirector works

For each port configured on the Redirector, an administrator has the following security options:

- Pass-through - data between the client and the host is not altered
- Client side - encrypts data between the client and the redirector
- Host side - encrypts data between the redirector and the host
- Both - does client-side and host-side security

You must create the HODServerKeyDb.kdb for the Redirector before you can enable client-side security, server-side security, or both.

You can use pass-through when encryption by the Redirector is not necessary, either because the data stream does not need to be encrypted, or because the data stream is already encrypted between the client and the host. You must use pass-through if the Host On-Demand client is connecting through the Redirector to a host that requires client authentication or Express Logon.

Refer to adding a host to the Redirector in the online help for more information.

Using Host On-Demand with a firewall

If you are configuring Host On-Demand to go through a firewall, we recommend that the firewall administrator open only those ports required for the clients to function. Telnet ports allow TLS or SSL-encrypted session traffic.

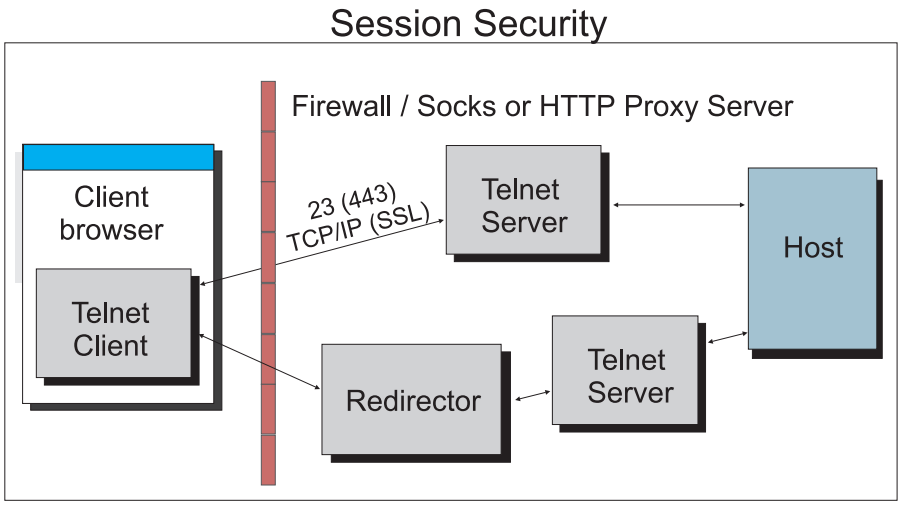

Figure 6. Session security through a firewall or proxy server

The Host On-Demand configuration servlet allows Host On-Demand clients to communicate with the configuration server across either HTTP or HTTPS.

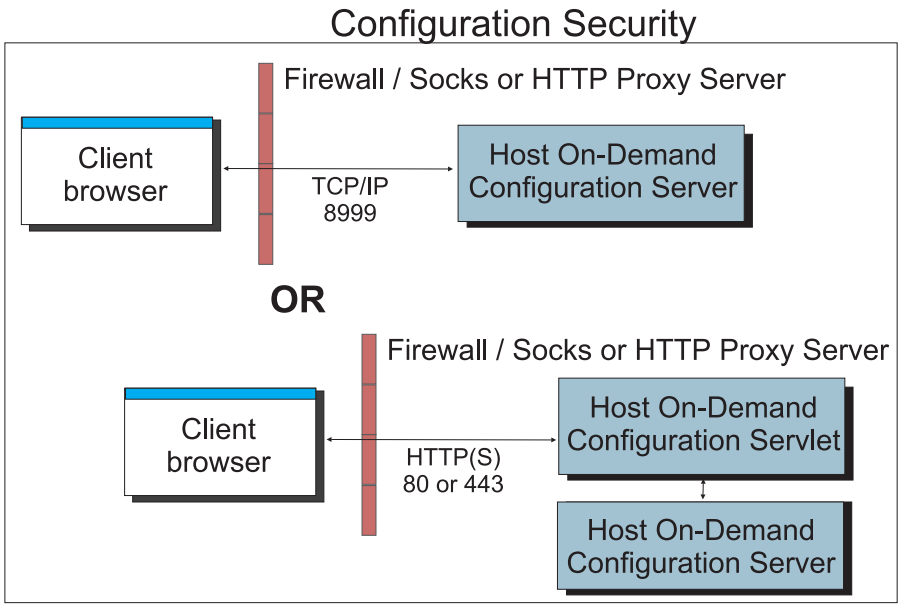

Figure 7. Configuration security with and without the configuration servlet through a firewall or proxy server

Host On-Demand clients connecting to a host system through open ports in the firewall should see "Configuring firewall ports" for details. Host On-Demand clients connecting to a host system through a Socks or HTTP proxy server should see "Connecting to a host system through a proxy server" on page 50 for details.

Configuring firewall ports

If you are using the configuration server-based model or the combined model, your Host On-Demand clients will need to communicate with the configuration server. To allow this through a firewall, you will need to either open the Host On-Demand Service Manager port or use the Host On-Demand configuration servlet. The Service Manager listens on port 8999 by default. You can change this default to any other available port number. For details, refer to Changing the Service Manager port in the online help. The Host On-Demand configuration servlet allows Host On-Demand clients to communicate with the configuration server across either HTTP or HTTPS. Therefore, the Service Manager port does not need to be open on the firewall. (See Figure 4 on page 25.) Refer to "Installing the configuration servlet" on page 69 and Configuring the configuration servlet in the online help for details on using the configuration servlet.

If you are using the HTML-based model, there is no requirement for Host On-Demand clients to access the configuration server, and the Service Manager port does not need to be open on the firewall. The clients will still attempt to contact the configuration server for license counting but will fail silently if the Service Manager port is not open. If you want to prevent clients from making license counting requests, you can add a parameter Disable with a value of LUM in the Additional Parameters tree view on the Advanced Options window in the Deployment Wizard.

In addition to the Service Manager port, make sure the firewall administrator opens any ports that are being used for functions your clients use. For example, if

you have a TLS or SSL session with the Redirector on port 5000, port 5000 must be open for Telnet traffic. The following table summarizes the ports that Host On-Demand can use.

Table 12. Host On-Demand functions and the ports they use

Host On-Demand Function	Ports Used
Display emulation (3270 and VT) and 3270 Printer emulation	23 (Telnet), 80 (HTTP), or 443 (TLS or SSL) and 8999 (config server)[3]
5250 Display and Printer emulation	23 (Telnet) or 992 [1] (TLS or SSL) or 80 (HTTP) or 443 (TLS or SSL) and 8999 (config server) [3]
3270 file transfer	23 (Telnet), 80 (HTTP), or 443 (TLS or SSL) and 8999 (config server)[3]
5250 file transfer - savfile	80 (HTTP), 8999 (config server)[3], 21 (FTP)[4], >1024 (FTP)[4], 446 (drda)[4], 449 (as-svrmap)[4], 8470 (as-central)[1 2 4], 8473 (as-file)[1 4], 8475 (as-rmtcmd)[1 4], and 8476 (as-signon)[1 4]
5250 file transfer - database	80 (HTTP), 8999 (config server)[3], 446 (drda)[4], 449 (as-svrmap)[4], 8470 (as-central)[1 2 4], 8473 (as-file)[1 4], 8475 (as-rmtcmd)[1 4], and 8476 (as-signon)[1 4]
5250 file transfer - stream file	80 (HTTP), 8999 (config server)[1 2 4], 449 (as-svrmap)[4], 8470 (as-central)[1 2 4], 8473 (as-file)[1 4], and 8476 (as-signon)[1 4]
FTP	21 (FTP), 80 (HTTP), 8999 (config server)[1 2 4], and >1024 (FTP)[5]
CICS	2006
Database On-Demand	80 (HTTP), 8999 (config server)[3], 449 (as-svrmap)[4], 8470 (as-central)[1 2 4], 8471 (as-database)[1 4], and 8476 (as-signon)[1 4]
License Use Management (LUM)	8999 (config server) for default license use counting using the configuration server
Host On-Demand clients	23 (Telnet), 80 (HTTP), and 8999 (config server)[3]
Administration clients	80 (HTTP) and 8999 (config server)[3]
SSH (the Secure Shell)	22

Notes:

1 You can change the port numbers with the command `WRKSRVTBLE` . The port numbers listed are the default values.

2 The port for as-central is used only if a codepage conversion table needs to be created dynamically (EBCDIC to/from Unicode). This is dependant on the JVM and the locale of the client.

3 You can change the config server port. Port 8999 is the default.

4 These ports do not need to be opened on the firewall if you are using iSeries proxy server support. You will need to open the default proxy server port 3470. You can change this port.

5 In passive (PASV) mode, the FTP client initiates both connections to the server, solving the problem of firewalls filtering the incoming data port connection to the client from the server. When opening a FTP connection, the client opens two random unprivileged ports locally (N>1024 and N+1). The first port contacts the server on port 21, but instead of then issuing a PORT command and allowing the server to connect back to its data port, the client issues the PASV command. As a result, the server then opens a random unprivileged port (P>1024) and sends the PORT P command back to the client. The client then initiates the connection from port N+1 to port P on the server to transfer data.

From the server-side firewall's standpoint, to support passive mode FTP, you must open the following communications ports:

- FTP server's port 21 from anywhere (client initiates connection)
- FTP server's port 21 to remote ports >1024 (server responds to client's control port)
- FTP server's ports >1024 from anywhere (client initiates data connection to random port specified by server)
- FTP server's ports >1024 to remote ports >1024 (server sends ACKs (and data) to client's data port)

If you do not want to open port 8999 on the firewall, you can still allow users to access Host On-Demand. There are two options:

- Use the Deployment Wizard to create HTML files that contain all configuration information. This eliminates the need to access the configuration server. When creating the HTML files, choose "HTML-based model" from the Configuration Model page of the Deployment Wizard.
- If you want to use the configuration server, you can configure clients to use the configuration servlet. Refer to Configuring the configuration servlet in the Host On-Demand online help. This option is only available if your Web server supports servlets.

  If you use the configuration server and it is separated from your Web browser by a firewall, you will either need to open the configuration server port on the firewall or run the Host On-Demand configuration servlet. The configuration servlet allows the browser to communicate with the configuration server across standard Web protocols, such as HTTP or HTTPS. (See Figure 4 on page 25.)

Connecting to a host system through a proxy server

Host On-Demand clients can use a proxy server to transparently access host systems from behind a firewall. Two types of proxy servers are supported:

- Socks proxy servers, described in "Connecting through a Socks proxy server" on page 51. Both version 4 and version 5 of Socks are supported.
- HTTP proxy servers, described in "Connecting through an HTTP proxy server" on page 51.

Before you can connect to a host system through a proxy server, you must find out which protocol the proxy server supports. Decide whether you want to specify the proxy server settings through the Web browser or explicitly identify a proxy server for the session. If you decide to explicitly identify a proxy server, you must specify the protocol that the proxy server uses, the proxy server name and port number, and other information.

In general, if a Socks proxy server is available, configure Host On-Demand sessions to use it. Configure sessions to use an HTTP proxy server if that is the only type of proxy server supported at your site.

Connecting through a Socks proxy server

Many organizations use Socks proxy servers to protect computing resources behind a firewall. Socks is a protocol for TCP/IP-based network proxies. It allows applications on one side of a Socks proxy server to gain full access to hosts on the other side of the Socks proxy server without directly connecting to them. Proxy servers are generally used in conjunction with firewalls. Under the Socks protocol, a client that requests a connection to a host system through a firewall actually connects to a Socks proxy server. The Socks proxy server acts as an intermediary between the client and the host system. It authorizes communication requests, connects to the host on behalf of the client, and relays data between the two systems.

Host On-Demand supports both version 4 and version 5 of the Socks protocol.

- Socks version 4 specifies the message format and conventions to allow TCP-based application users access across a firewall. It provides access control based on TCP header information, including IP addresses and source and destination port numbers.
- Socks version 5 (also known as authenticated firewall traversal (AFT)) is an open Internet standard for network proxies. It adds authentication, better support for resolving domain names, support for IPv6 addresses, and other features to version 4. These features are very useful for clients located outside a firewall. A Socks user ID and password for the proxy server can optionally be sent over the connection between the Host On-Demand client and the proxy server. The user ID and password are not encrypted. For more information on version 5, see *Socks Protocol Version 5* (RFC 1928), available at http://www.ietf.org/rfc/rfc1928.txt?number=1928.

The Java Virtual Machine (JVM) used in most Web browsers supports Socks version 4. A session can access either a Socks version 4 or version 5 proxy server, bypassing the proxy server settings in the Web browser. You can also have the session negotiate a Socks version 4 connection if the proxy server does not support version 5. For more information on Socks proxy server settings, refer to Proxy Server in the online help.

Connecting through an HTTP proxy server

HTTP proxy servers handle HTTP requests through firewalls. They act as intermediaries between private local networks and the Internet. The HTTP proxy server is connected to both the local network and the Internet. Local users configure their browsers to pass HTTP requests through the HTTP proxy server by specifying the proxy server's IP address and TCP port number. The HTTP proxy server accepts these HTTP requests and forwards them to the actual Web servers specified by the URLs entered in the browser.

For Host On-Demand clients, HTTP proxy servers act as forwarding agents for connections to a host system. The HTTP proxy server opens a connection to the host system and sends data back and forth between the host system and the client. Although an HTTP proxy server usually closes a connection after servicing an HTTP request, Host On-Demand keeps the connection open for host traffic by using the HTTP Connect method (if it is enabled for the proxy server).

To have a session use a HTTP proxy server, you need to select HTTP proxy as the proxy type and specify the proxy server name and port number. For more information on HTTP proxy server settings, refer to Proxy Server in the online help.

User ID security

Web Express Logon

If you have a network security application in place and you are using the configuration server-based model, you can select Web Express Logon in the Deployment Wizard to allow users to access hosts and host-based applications without providing an additional user ID and password. Entering the full URL of the Credential Mapper Server tells Host On-Demand where to locate the Credential Mapper Servlet, which processes the HTTPS request from the user, performs a lookup, and returns the user's credentials. The credentials are then used to perform a secure, automated Host On-Demand login.

Native Authentication

If you use the configuration server-based model, you can configure your Host On-Demand users to be natively authenticated. This option allows users to log on to Host On-Demand using the same password as they would to log on to the operating system (Windows NT, AIX, or z/OS) where Host On-Demand is active. When a user logs on to Host On-Demand, their password is validated against the operating system password, rather than a separate Host On-Demand password. This gives the administrator a single point of control for password administration and the user a single password to remember.

Refer to Native Authentication in the online help for more information on enabling this option.

Windows Domain logon

If your users are logged on to a Windows domain, this option (available with the configuration server-based model in the Deployment Wizard) automatically logs users on to Host On-Demand using their Windows user name. The Host On-Demand logon window does not appear and the Windows user name is used as the Host On-Demand user ID. If a Host On-Demand user ID does not already exist (matching the Windows user name), you can also choose to have a user ID automatically created in the specified Host On-Demand group.

Refer to Logon Type in the online help for more information on choosing how users access the Host On-Demand configuration server.

Chapter 6. Planning for national language support

Host On-Demand is provided in 23 languages. The session windows, configuration panels, help files, and the documentation have been translated. In addition, display, keyboard, and processing support is provided for Arabic, Hebrew, Thai, and Hindi. This support is fully explained in the online help.

All the translated versions are provided on the CDs and on the zSeries tapes. When you install Host On-Demand on OS/400, Windows, AIX, Linux, Solaris, and HP-UX using the graphical installation program, you can choose which languages to install. On z/OS, OS/2, and Novell, all the languages are always installed.

National language support is operating-system dependent, so the appropriate font and keyboard support for the language you want to use must be installed in the operating system. For example, if you want to use French as the host-session language but do not have the French font and keyboard support installed, you may not be able to display the correct characters.

DBCS cannot be used as the HTML file name.

Supported languages

The languages into which Host On-Demand has been translated are listed below, along with the language suffixes you can use to load translated versions of the Host On-Demand clients. For example, IBM-supplied HTML pages have language extensions to identify different language installations and different language predefined HTML files.

Language	Language suffix
Simplified Chinese	zh
Traditional Chinese	zh_TW
Czech	cs
Danish	da
Dutch	nl
English	en
Finnish	fi
French	fr
German	de
Greek	el
Hungarian	hu
Italian	it
Japanese	ja
Korean	ko
Norwegian	no
Polish	pl

Brazilian Portuguese	pt
Portuguese	pt_PT
Russian	ru
Slovenian	sl
Spanish	es
Swedish	sv
Turkish	tr

Supported host code pages

Host On-Demand supports multiple code pages. You can specify these code pages on a session-by-session basis.

3270 and 5250 code pages

The code pages specified below are supported by the 3270 and 5250 emulators. You can select them in the Session Configuration window.

Country or region	Code page	Note
Arabic Speaking	420	
Austria	273	
Austria (Euro)	1141	
Belarus	1025	
Belarus (Euro)	1154	
Belgium	037	
Belgium (Euro)	1140	
Belgium (Old Code)	274	
Bosnia/Herzegovina	870	
Bosnia/Herzegovina (Euro)	1153	
Brazil	037	
Brazil (Euro)	1140	
Brazil (Old)	275	
Bulgaria	1025	
Bulgaria (Euro)	1154	
Canada	037	
Canada (Euro)	1140	
China (Simplified Chinese Extended)	1388	
Croatia	870	
Croatia (Euro)	1153	
Czech Republic	870	
Czech Republic (Euro)	1153	
Denmark	277	
Denmark (Euro)	1142	
Estonia	1122	

Estonia (Euro)	1157	
Finland	278	
Finland (Euro)	1143	
France	297	
France (Euro)	1147	
FYR Macedonia	1025	
FYR Macedonia (Euro)	1154	
Germany	273	
Germany (Euro)	1141	
Greece	875	
Hebrew (New Code)	424	
Hebrew (Old Code)	803	
Hindi	1137	5250 display only
Hungary	870	
Hungary (Euro)	1153	
Iceland	871	
Iceland (Euro)	1149	
Italy	280	
Italy (Euro)	1144	
Japan (Katakana)	930	
Japan (Katakana Extended)	930	
Japan (Katakana Unicode Extended)	1390	3270 only
Japan (Latin Extended)	939	
Japan (Latin Unicode Extended)	1399	
Korea (Euro)	1364	
Korea (Extended)	933	
Latin America	284	
Latin America (Euro)	1145	
Latvia	1112	
Latvia (Euro)	1156	
Lithuania	1112	
Lithuania (Euro)	1156	
Multilingual	500	
Multilingual ISO (Euro)	924	
Multilingual (Euro)	1148	
Netherlands	037	
Netherlands (Euro)	1140	
Norway	277	
Norway (Euro)	1142	
Open Edition	1047	

Poland	870	
Poland (Euro)	1153	
Portugal	037	
Portugal (Euro)	1140	
Romania	870	
Romania (Euro)	1153	
Russia	1025	
Russia (Euro)	1154	
Serbia/Montenegro (Cyrillic)	1025	
Serbia/Montenegro (Cyrillic; Euro)	1154	
Slovakia	870	
Slovakia (Euro)	1153	
Slovenia	870	
Slovenia (Euro)	1153	
Spain	284	
Spain (Euro)	1145	
Sweden	278	
Sweden (Euro)	1143	
Taiwan (Traditional Chinese Extended)	937	
Taiwan (Traditional Chinese Extended; Euro)	1371	
Thai	838	
Thai (Euro)	1160	
Turkey	1026	
Turkey (Euro)	1155	
Ukraine	1123	
Ukraine (Euro)	1158	
United Kingdom	285	
United Kingdom (Euro)	1146	
United States	037	
United States (Euro)	1140	

Notes:

- 3270 host print with a Printer Definition Table (PDT) supports only Latin-1, DBCS, bidirectional, and Thai code pages. Other code pages are supported either in Adobe PDF printing or on Windows platforms without a PDT.
- In order to include more characters (which are defined in the GB18030 standard by the Government of the People's Republic of China), 6582 Unicode Extension-A and 1,948 additional non-Han characters (Mongolian, Uygur, Tibetan, and Yi) were added to the Simplified Chinese code page 1388 for Host On-Demand Version 6.

VT code pages

Language	Code page
Arabic	ASMO 708 and ASMO 449
British	1101
DEC Greek	
DEC Hebrew	
DEC Multinational Replacement Character Set	1100
DEC Technical	
Dutch	1102
Finnish	1103
French	1104
French Canadian	1020
German	1011
Hebrew NRCS	
ISO Greek Supplemental (ISO Latin-7)	813
ISO Hebrew Supplemental	
ISO Latin-1	819
Italian	1012
Norwegian/Danish	1105
PC Danish/Norwegian	865
PC International	437
PC Multilingual	850
PC Portugese	860
PC Spanish	220
Spanish	1023
Swedish	1106
Swiss	1021
United States	1100

CICS Gateway code pages

Code page	Character set
000	Auto Detect (default)
437	Latin-1
813	ISO Greek (8859_7)
819	ISO Latin 1 (8859_1)
850	Latin 1
852	Latin 2
855	Cyrillic
856	Hebrew
857	Latin 5

864	Arabic
866	Cyrillic
869	Greek
874	Thai
912	ISO Latin 2 (8859_2)
915	ISO Cyrillic (8859_5)
920	ISO Latin 5 (8859_9)

User-defined character mapping

For double-byte character set (DBCS) languages, you can use customized user-defined character (UDC) mapping in your session (3270, 5250, 3270 host print) instead of the default mapping. You can create a UDC translation table using the UDC mapping editor to store customized mapping for your session. For instructions for how to use the UDC mapping editor to change your character mapping, see Using the user-defined character (UDC) mapping editor in the online help.

Unicode Support for OS/400

See “Unicode Support for OS/400” on page 153.

Part 2. Installing, upgrading, and uninstalling Host On-Demand

Chapter 7. Installing the Host On-Demand server and related software

This chapter discusses the installation of the following three Host On-Demand components:

- The Host On-Demand server, which is necessary for using Host On-Demand. Refer to "Installing the Host On-Demand server" for instructions.
- The Host On-Demand configuration servlet, which is needed only in specific instances when you are running Host On-Demand in conjunction with a firewall. Refer to "Installing the configuration servlet" on page 69 for further explanation and instructions.
- The Deployment Wizard, an extremely useful tool that runs on Windows to generate customized Host On-Demand clients. Installing the Deployment Wizard is not required, but it is highly recommended. Refer to "Installing the Deployment Wizard" on page 70 for instructions.

If you are upgrading to Host On-Demand 8 from a previous version, refer to Chapter 8, "Upgrading from earlier versions of Host On-Demand", on page 71 for instructions on how to upgrade your system.

Installing the Host On-Demand server

Before installing the Host On-Demand server, ensure that you have the appropriate level of authority to access the directories and run the commands required for installation. For example:

- On Windows, you must log in as Administrator or as a user that is a member of the Administrators group.
- On OS/400, you must sign on with the QSECOFR user profile (or with another user profile with equivalent security authorities).
- On any Unix-based operating system, you must log on with root access authority.

Because Host On-Demand clients are served as Web pages, you must install the server component in the same environment as a Web server.

Installing on z/OS or OS/390

If you are upgrading from a previous version of Host On-Demand, refer to Chapter 8, "Upgrading from earlier versions of Host On-Demand", on page 71 for information on backing up your customized HTML pages and other customized configuration files.

For instructions about installing Host On-Demand on z/OS or OS/390, refer to the Host On-Demand Program Directory supplied with the Host On-Demand product media.

For instructions on installing Host On-Demand on the Linux/390 operating system, refer to "Installing on Windows, AIX, Linux, Solaris, and HP-UX" on page 64.

For information on configuring Host On-Demand on zSeries servers, refer to Chapter 16, “Configuring Host On-Demand on zSeries”, on page 137.

Installing on OS/400

There are three options for installing the Host On-Demand server on OS/400 systems:

- “Using the graphical interface for remote installation”
- “Using the console or silent mode for local installation” on page 63
- “Running a remote console or silent installation from a Windows machine” on page 64

Using the graphical interface for remote installation

To install on OS/400 in graphical mode, you must install remotely from a computer running Windows. The following steps guide you through the install:

1. Insert the Host On-Demand CD into your Windows system and open the Windows command prompt window. If your computer has CD autoplay, exit out of the Host On-Demand Welcome window.
2. At the command line prompt, change to the hodinst directory and enter the Windows launcher with an additional parameter specifying the OS/400 operating system:

   ```
   hodinstallwin.exe -os400
   ```

   Alternatively, you can use three more parameters to designate the exact server to which you are installing and log onto that server. For example:

   ```
   hodinstallwin.exe -os400 myserver myuserid mypassword
   ```

   Myserver is the TCP/IP address or host name for your iSeries server. *Myuserid* and *mypassword* are a valid logon ID to that server.
3. If you do not specify the iSeries server and your logon ID in the command line, a window appears prompting you to enter that information. After you enter that information, the wizard starts. It automatically uses the language of your location, defined on your system by the running Java Virtual Machine (JVM).
4. Read the software agreement. You must accept the software agreement to continue the installation.
5. If you have a previous version of Host On-Demand installed, a window appears instructing you to uninstall it. For Host On-Demand 4, 5, 6, and 7, click Next to automatically delete the previous version. All of your existing customized HTML files and other customized configuration files will be saved.

   After the previous version is deleted, Host On-Demand 8 installation continues. Customized files will be restored after Host On-Demand 8 installation completes.
6. The additional language selection window appears to allow you to choose support for multiple languages in addition to English, which is automatically installed.
7. A list of Web servers detected on the iSeries system appears. Select which Web server you want to configure for Host On-Demand. For a list of supported Web servers, refer to “Web servers” on page 18.
8. Specify the Service Manager port, through which Host On-Demand clients communicate with the Service Manager. This communication is necessary for the following deployment options:

- Using the configuration server to maintain session configuration information (as in the configuration server-based and combined deployment models, described in Chapter 3, “Planning for deployment”, on page 23)
- License-Use Counting (refer to License Usage in the online help)

IBM recommends designating port 8999 for these purposes. Check your server documentation to see if this port is being used. If it is in use, you can change the port during installation or at a later time. For more information about changing the Service Manager port, see Changing the Service Manager’s configuration port in the online help.

9. If the installation program detects IBM WebSphere Application Server (versions 4.0 and 5.0) on your system, the next window asks if you want to configure the Host On-Demand configuration servlet in WebSphere Application Server. If you run Host On-Demand through a firewall, this eliminates the need to open an extra port for client communications with the Host On-Demand Service Manager. See “Installing the configuration servlet” on page 69 for more information.
   - If you click Yes, a window appears listing the versions of the application servers detected, prompting you to choose from them. The installation program automatically deploys the configuration servlet on the Web application server you designate, and it configures your clients to access the Service Manager through the servlet.
   - If you click No, the install configures the clients to access the Service Manager directly on port 8999 (or an alternative port you have specified).
10. A window summarizing all of your input appears. Review and click Next to install.
11. When you see the installation complete message, click Finish to exit the wizard.

Using the console or silent mode for local installation

Installing Host On-Demand in console mode suppresses the GUI wizard. Instead, the utility sends messages and text prompts directly to your console (or command line window). You make selections by pressing the Enter key or typing a number.

The silent mode is particularly useful for deploying multiple images of Host On-Demand server. The silent mode requires no interaction between you and the systems constituting your installation. You simply distribute a text-only response file supplying installation input.

The following steps apply to both console and silent installations on your iSeries server:

1. Place the Host On-Demand installation CD in the CD-ROM drive of your iSeries server.
2. Sign on with the `QSECOFR` user profile or a profile with equivalent security authorities.
3. Enter `STRQSH` at the command line to start the Qshell interpreter.
4. Enter `cd /QOPT/HOD/instmgr` to change directories to the installation CD’s `instmgr` directory.
5. Run the following shell script according to your installation mode:
   - Console: `inst400.sh`
   - Silent: `inst400.sh -silent -options/mydirectory/responseFile`

For other installation options, refer to Appendix D, “Native platform launcher command line options”, on page 183.

Running a remote console or silent installation from a Windows machine

To run a remote console installation from a Windows machine, enter the following: `hodinstallwin.exe -os400 -console`.

To run a remote silent installation from a Windows machine, enter the following:

```
hodinstallwin.exe -os400 myserver myuserid mypassword -silent -options
c:\mydirectory\responseFile
```

Installing on Windows, AIX, Linux, Solaris, and HP-UX

There are three options for installing the Host On-Demand server on Windows, AIX, Linux, Solaris, and HP-UX systems:

- “Using the graphical interface”
- “Using the console mode” on page 66
- “Using the silent mode” on page 66

Even if you plan to install in console or silent mode, you should read through the steps for using the graphical interface. They document environment variables required for any installation mode.

Using the graphical interface

The following steps guide you through the graphical interface for installation on Windows, AIX, Linux, Solaris, and HP-UX:

1. If your platform supports CD autoplay, insert the CD and wait for the start window. If not, you must launch the installation program with the native platform launcher appropriate to your environment. Use one of the following (on the CD in the `hodinst` directory):
   - `hodinstallwin.exe` for Windows
   - `hodinstallwin.console.exe` for Windows. Launches the Windows console with return codes.
   - `hodinstall_aix.bin` for AIX
   - `hodinstall_linux390.bin` for Linux/390
   - `hodinstall_linuxppc.bin` for Linux partitions on pSeries and iSeries
   - `hodinstall_linux.bin` for all other Linux versions
   - `hodinstall_solaris.bin` for Solaris
   - `hodinstall_hpux11x.bin` for HP-UX

   As you enter your native platform launcher, you can add command-line parameters to the installation process. Refer to Appendix D, “Native platform launcher command line options”, on page 183 for more information.
2. The welcome window appears in the language of your system or user locale.
3. Read the software agreement, which you must accept to continue installation.
4. If you have a previous version of Host On-Demand installed, the next window instructs you to uninstall it.

   For Host On-Demand 5, 6, or 7 on Windows and AIX, click Next to automatically launch the uninstall utility. It performs the uninstallation, leaving all of your existing customized HTML pages and other customized configuration files in place.

   When the utility closes, Host On-Demand 8 installation continues.

5. Next the wizard prompts you for the installation directory. If you are upgrading from a previous version of Host On-Demand, your previous installation directory appears as the default on Windows and AIX only. Otherwise, the installation directory defaults to one of the following:
   - `c:\Program Files\IBM\HostOnDemand` for Windows
   - `/opt/IBM/HostOnDemand` for AIX, Linux, Solaris, and HP-UX
6. The additional language selection window appears to allow you to choose support for multiple languages in addition to English, which is automatically installed.
7. The next window asks for input to configure appropriate Web servers and establish the publish directory.
   a. A list of detected Web servers appears. Select which Web server you want to configure for Host On-Demand. For a list of supported Web servers, refer to "Web servers" on page 18.
   b. The publish directory stores files that must be kept available to clients. The install wizard prompts you to designate your publish directory by displaying the default, `HOD`, as a subdirectory appended to your Host On-Demand server path. The wizard also prompts you to specify an alias for the directory.
8. Specify the Service Manager port, through which Host On-Demand clients communicate with the Service Manager. This communication is necessary for the following deployment options:
   - Using the configuration server to maintain session configuration information (as in the configuration server-based and combined deployment models, described in Chapter 3, "Planning for deployment", on page 23)
   - License-Use Counting (refer to License Usage in the online help)

   IBM recommends designating port 8999 for these purposes. Check your server documentation to see if this port is being used. If it is in use, you can change the port during installation or at a later time. For more information about changing the Service Manager port, see Changing the Service Manager's configuration port in the online help.
9. If the installation program detects IBM WebSphere Application Server (versions 4.0 and 5.0) on your system, the next window asks if you want to configure the Host On-Demand configuration servlet in one of them. If you run Host On-Demand through a firewall, this eliminates the need to open an extra port for client communications with the Host On-Demand Service Manager. See "Installing the configuration servlet" on page 69 for more information.
   - If you click Yes, a window appears listing the versions of the application servers detected, prompting you to choose from them. The installation program automatically deploys the configuration servlet on the Web application server you designate, and it configures your clients to access the Service Manager through the servlet.
   - If you click No, the install configures the clients to access the Service Manager directly on port 8999 (or an alternative port you have specified).

   If you click Yes and select WebSphere Application Server 5, and you have multiple servers configured within WebSphere Application Server, the install wizard prompts you to choose the server on which you want to deploy the configuration servlet.
10. A window summarizing all of your input appears. Click Next to install.

11. When the installation is complete, the wizard presents you with options to register your software and view the Host On-Demand InfoCenter.
12. When you click Finish in the next window, the wizard might prompt you to restart your computer.

For AIX administrators: Be aware that installing Host On-Demand 8 on AIX results in new compiler requirements for Certificate Management. Refer to Appendix B, "Using the IKEYCMD command-line interface", on page 167 for more information.

Using the console mode

Installing Host On-Demand in console mode suppresses the GUI wizard. Instead, the utility sends messages and text prompts directly to your console (or command line window). You make selections by pressing the Enter key or typing a number.

To use console mode, input your native platform launcher with the `-console` command line option. For example, on Windows:

```
hodinstallwin.exe -console
```

For other installation options, refer to Appendix D, "Native platform launcher command line options", on page 183.

Using the silent mode

The silent mode is particularly useful for deploying multiple images of Host On-Demand server. The silent mode requires no interaction between you and the systems constituting your installation. You simply distribute a text-only response file supplying installation input.

You can find a sample response file on the Host On-Demand CD in hodinst\hodSampleResponse.txt. After modifying the file for your environment, enter the following command-line options (with your native platform launcher) to run a silent installation. For example, on Windows:

```
hodinstallwin.exe -silent -options c:\mydirectory\responseFile
```

where `c:\mydirectory\responseFile` is your response file's path name.

Note: The directory and file name must already exist.

To create your own response file, enter the following options:

```
-options-record filename
```

where filename is the name of your response file.

For other installation options, refer to Appendix D, "Native platform launcher command line options", on page 183.

Installing on OS/2

If you are upgrading and have changed `/hostondemand/private/NSMprop` or changed or created `/hostondemand/hod/config.properties`, you must back up these files before installation and then restore them after installation. These files will be overwritten during the unzip process.

The following steps assume that hostondemand is the server directory and HOD is the publish directory. To install the Host On-Demand server:

1. Insert the CD.
2. Create a server directory, for example, hostondemand. The server directory contains files that are used only by the server and must not be available to client workstations.
3. Change to the server directory.
4. Run the following command to extract the files:

   ```
   unzip [cd_rom]:\zip\hod80srv.zip
   ```

   where:

   - *unzip* is your unpacking program (such as UNZIP.EXE). It must support long file names
   - *[cd_rom]* is the CD-ROM drive letter
   - *zip* is the directory on the CD
5. Create the publish directory; for example, HOD. The publish directory contains files that must be available to client users who access the server through a browser.
6. Change to the publish directory.
7. Run the following command to extract the files:

   ```
   unzip [cd_rom]:\zip\hod80www.zip
   ```
8. Make the publish directory available to clients on the network. Refer to your Web server documentation for information on how to do this.
9. Configure a local host by adding the following line to the setup.cmd file, which is usually found in the \mptn\bin directory:

   ```
   ifconfig lo 127.0.0.1
   ```
10. Start the Host On-Demand Service Manager, which provides support services for Host On-Demand and runs as a Java application:
    a. At the command prompt, change directory to \hostondemand\lib .
    b. Copy NCServiceManager-OS2.cmd from the \hostondemand\lib\samples\CommandFiles directory.
    c. Edit NCServiceManager-OS2.cmd to reflect the directory paths appropriate for your workstation.
    d. Run NCServiceManager-OS2.cmd. The Service Manager does not display a message indicating that it has started. Also, disregard the following message: *Native library failed to load, indicating this Redirector does not support SSL.* The failure to load this library simply indicates that the server does not support SSL sessions.

For Host On-Demand to function, the Service Manager must be running. If you reboot the server, you must also restart the Service Manager. You might want to add the NCServiceManager-OS2.cmd command to your startup.cmd file so that the Service Manager starts automatically when the workstation boots. If you do, remember to specify the path to change directory to the \hostondemand\lib subdirectory before the command runs.

11. Restart the Web server.
12. Now that your installation is complete, see Part 3, “Configuring Host On-Demand”, on page 81.

Installing on Novell NetWare

If you are upgrading and have changed /hostondemand/private/NSMprop or changed or created /hostondemand/hod/config.properties, you must back up these files before installation and then restore them after installation. These files will be overwritten during the unzip process.

These steps assume that hostondemand is the server directory and HOD is the publish directory. To install the Host On-Demand server:

1. Stop the Service Manager with the java -exit command.
2. From a client workstation, map a drive to the SYS: volume of the Novell server.
3. Insert the CD.
4. Create a server directory, for example, hostondemand. The server directory contains files that are only used by the server and must not be available to client workstations.
5. Change to the server directory.
6. From the drive mapped to the SYS: volume, run the following command to extract the files:

   ```
   unzip [cd_rom]:\zip\hod80srv.zip
   ```

   where:

   - *unzip* is your unpacking program (such as WinZip). It must support long file names.
   - *[cd_rom]* is the CD-ROM drive letter.
   - *zip* is the directory on the CD.
7. Create a publish directory named HOD and change to that directory. The HOD directory contains files that must be available to client users who access the Host On-Demand server through a browser.
8. Run the following command to extract the files:

   ```
   unzip [cd_rom]:\zip\hod80www.zip
   ```
9. From the server console, run the command load java to start the Java NLM.
10. Start the Host On-Demand Service Manager, which provides support services for Host On-Demand and runs as a Java application, by following these steps from a client system mapped to the SYS volume of the server:
    a. Copy NCServiceManager-Novell.ncf from the \hostondemand\lib\samples\CommandFiles directory to the \system directory on the Novell server. To run the command from the server console, you might have to change the file name to the eight-dot-three format.
    b. Edit NCServiceManager-Novell.ncf (or the eight-dot-three format of the file) to reflect the directory paths that are correct for your workstation.
    c. From the server, run NCServiceManager-Novell.ncf (or the eight-dot-three format of the file). The Service Manager does not display a message indicating that it has started.

For Host On-Demand to function, the Service Manager must be running. If you reboot the server, you must also restart the Service Manager.

11. Now that your installation is complete, see Part 3, “Configuring Host On-Demand”, on page 81.

Installing the configuration servlet

During the Host On-Demand installation, you can choose to have the configuration servlet installed and configured on OS/400, Windows, AIX, Linux, Solaris, and HP-UX for IBM WebSphere Application Server Version 4.0 and 5.0.

All Web servers and servlet engines are configured differently. Check your Web server and servlet engine documentation for servlet configuration details on your operating system.

Installing the configuration servlet is necessary only if both of the following statements are true for your Host On-Demand deployment:

- You plan to configure Host On-Demand so that client communication with the Service Manager is necessary (as in the configuration server-based and combined deployment models, if you enable License-Use Counting, or if you use the Redirector).
- A firewall protects the server(s) on which you plan to maintain session configuration information, and you do not want to open a port in that firewall to give outside clients access to the Service Manager.

By default, the Host On-Demand clients use port 8999 to access configuration information from the Service Manager. If any of your clients are outside the firewall, the firewall administrator needs to open port 8999 both internally and externally. However, you can avoid opening this port by customizing your clients to use the configuration servlet to access configuration information.

Deploying the servlet on WebSphere Application Server

During Host On-Demand installation on Windows, AIX, Linux, Solaris, and HP-UX, the install utility searches your system for an instance of WebSphere Application Server. If it detects an instance, the install utility can automatically install and configure the configuration servlet on WebSphere Application Server versions 4.0 and 5.0.

If you need to manually install the configuration servlet, look in your WebSphere Application Server documentation for steps on installing enterprise applications. You can also go to `http://www.ibm.com/software/webservers/` and navigate to the WebSphere Application Server support page, where you will find a link to your version’s InfoCenter.

The Host On-Demand configuration servlet EAR file, cfgservlet.ear, is located in the lib directory of your Host On-Demand installation.

For WebSphere Application Server 5: After you save your deployment settings in the administrative console, you need to start the Host On-Demand configuration servlet in the Enterprise Applications window of WebSphere Application Server. Then go to the Environment window and select Update Web Server Plug-in.

After the configuration servlet is installed, you must configure your clients to use the configuration servlet instead of directly accessing the Service Manager. You can use the Deployment Wizard to build customized HTML client pages. The wizard sets the applet parameters in the HTML based on your input, so you do not have

to learn the syntax and valid parameter values. IBM recommends that you use the Deployment Wizard to set the ConfigServerURL parameter in the client HTML to `HODConfig/HODConfig/hod`.

For more information regarding configuration servlet parameters, configuration and examples, see Configuring the configuration servlet in the online help.

Installing the Deployment Wizard

The Deployment Wizard is automatically installed as part of the Windows Host On-Demand server installation. It is also available separately for those customers who do not wish to install the entire Windows Host On-Demand server. This separate Deployment Wizard can be installed in one of two ways:

- Using the Deployment Wizard install option on a Windows machine with the Host On-Demand CD.
- Downloading it from the Host On-Demand server.

The following two sections describe the installation process for each method.

The Deployment Wizard installation image is approximately 55 MB. If you plan to download this installation image, particularly over a modem, prepare for a large download.

Installing the Deployment Wizard from the Host On-Demand CD

To install and run the Deployment Wizard, do the following:

1. Insert the Host On-Demand CD. If autorun is enabled, the CD Installer starts automatically. If autorun is not enabled, start the CD Installer by running the setupwin.exe file located on the Host On-Demand CD.
2. From the CD Installer window, select Install Deployment Wizard.
3. A wizard guides you through the remaining installation steps.
4. Once installation is complete, you can launch the Deployment Wizard from the Start > Programs desktop menu.

Downloading the Deployment Wizard installation image from a Host On-Demand server

The Deployment Wizard image is shipped on all Host On-Demand server platforms, and it can be downloaded from the server and installed on any Windows machine.

To download the Deployment Wizard from a Host On-Demand server, do the following:

1. From your Windows machine, start your browser and point to the HODMain_*xx*.html file on your Host On-Demand server, where *xx* is your two letter language suffix.
2. Click on the Deployment Wizard link. This will download the Deployment Wizard installation image to your Windows machine.
3. Run the Deployment Wizard installation from your Windows machine.
4. Once installation is complete, you can launch the Deployment Wizard from the Start > Programs desktop menu.

Chapter 8. Upgrading from earlier versions of Host On-Demand

This chapter provides detailed information on how to properly upgrade your system from earlier versions of Host On-Demand. It discusses the steps involved in upgrading the following components of Host On-Demand:

Upgrading the Host On-Demand server

When upgrading the Host On-Demand server, the following basic steps minimize migration risks, and provide a transparent upgrade experience for your users:

1. Back up all of your customized Host On-Demand files (those making up your private directory, as well as modified or newly created files in the publish directory)
2. Perform the upgrade
3. Redeploy your customized files

After the entire migration process, users select from sessions with the same definitions as before. All of their customizations (for example, macros and keyboard remaps) continue to work as before.

The following sections guide you through these basic steps, which vary according to your operating system and Host On-Demand version upgrade.

Backing up files and directories

Upgrading on Windows or AIX: If your customized files include CustomizedCAs.class files generated by IKEYMAN (the Certificate Management utility built into Host On-Demand), be aware that upgrading to Host On-Demand 8 involves automatic translation of those files into a different format: CustomizedCAs.p12. For more information, refer to "Migrating from CustomizedCAs.class to CustomizedCAs.p12" on page 74.

If you need to modify the NSMprop file (for example, change the default port) before migration, or if you simply need to keep NSMprop intact during migration, put this file in the `/private` directory of your current version of Host On-Demand server.

Migration scenarios

IBM recommends different migration scenarios (including different file back-up methods), depending on your operating system and the Host On-Demand version from which you are upgrading to Host On-Demand 8.

Table 13. Migration scenarios

Operating system	Previous version of Host On-Demand	Migration scenario
• Windows: 98, NT 4.0, ME, 2000, XP • AIX: 4.3.3, 5.1, 5.2	5–7	Host On-Demand automatically uninstalls the previous version from your system and replaces it with Host On-Demand 8, leaving customized files intact. Refer to "Installing on Windows, AIX, Linux, Solaris, and HP-UX" on page 64.
Windows or AIX	previous to 5	Refer to "Migrating on server operating systems with an uninstall program" on page 74.
OS/400	4–7	Host On-Demand automatically uninstalls the previous version from your system and replaces it with Host On-Demand 8, leaving customized files intact. Refer to "Installing on Windows, AIX, Linux, Solaris, and HP-UX" on page 64.
Any other operating system without a native uninstall utility	does not apply	Refer to "Migrating on server operating systems without an uninstall program" on page 74.

Setting up a separate user publish directory

In Host On-Demand 7 and later, you can put custom HTML files (files generated from the Deployment Wizard), config.properties, and CustomizedCAs.class or CustomizedCAs.p12 files in a directory other than the Host On-Demand publish directory.

Creating a separate user publish directory makes it easier to apply Host On-Demand upgrades because installing a new version of Host On-Demand will not affect the new directory. It also keeps the Host On-Demand publish directory read-only because it provides a separate writeable location for deploying Deployment Wizard pages. Additionally, creating a separate user publish directory isolates customer generated files from those provided by Host On-Demand. Note that other user-modified files (such as customer applets and HACL programs) still need to run from the Host On-Demand publish directory.

1. To set up a separate user publish directory, do the following:
   - For the Download client or Cached client, specify the code base. The code base is the Host On-Demand server's publish directory, not the name of your new separate user publish directory:
     a. Using the Deployment Wizard, on the Additional Options window, click Advanced Options.
     b. Open the Code base window.
     c. Enter the code base. You can enter a fully qualified URL including the host name (for example, http://your_HOD_server/hod_publish_dir_alias/) or a relative path (for example, /hod_publish_dir_alias/).

Continue with step 2.

- For a Web Start Client, specify the document base. The Web Start client is an application, and therefore does not have a built-in method to determine where the HTML file is loaded. The document base allows you to specify the location of your HTML file. For more information about the document base, refer to Web Start Settings in the online help.

  Continue with step 2.

2. Select Output Zip to save the files generated from the Deployment Wizard in a Zip file.
3. Click Create File(s).
4. If you are not running the Deployment Wizard on your Host On-Demand server, FTP the output ZIP file to your server platform.
5. Create a separate user publish directory, /user_publish_dir/.
6. Use the DWunzip tool to install the Deployment Wizard generated files into the /user_publish_dir/ directory. You must edit the DWunzip command file on your server to specify the correct MY_PUBLISHED_DIRECTORY value. See the online help topic Using Dwunzip for more information on how to use this tool.

   The Deployment Wizard HTML files are installed in the directory /user_publish_dir/. Additional files like cfg0.cf and params.txt are installed in the /user_publish_dir/HODData/your_html directory.
7. Add a pass rule (also known as an alias on some platforms) in your Web server configuration file to point to this new user publish directory. For example, on IBM HTTP Server or Apache HTTP Server, add the following to /etc/httpd.conf:

   ```
   Pass  /user_alias/ *  /user_publish_dir/ *
   ```
8. If changes are required in the Host On-Demand config.properties file (for example, to change the default port or enable the Host On-Demand configuration servlet), do the following:
   a. Update the config.properties file. If your server platform does not support the ASCII character set, update this file on a machine that does support ASCII.
   b. If the config.properties file was updated on a different platform than your server, FTP the file to your server platform in binary format.
   c. Place the file in the user publish directory, /user_publish_dir/.
   d. Add a pass rule (also known as an alias on some platforms) in the Web server configuration file. For example, on IBM HTTP Server or Apache HTTP Server, add the following to /etc/httpd.conf:

      ```
      /hod_publish_dir_alias/config.properties
      /user_publish_dir/config.properties
      ```

      

      On the zSeries platform, append the ascii extension, /user_publish_dir/config.properties.ascii.
9. If you are using SSL and need to change the CustomizedCAs.12 file, do the following:
   a. Place the updated file in the user publish directory /user_publish_dir/CustomizedCAs.p12.
   b. Add a pass rule (also known as an alias on some platforms) in the Web server configuration file. For example, on IBM HTTP Server or Apache HTTP Server, add the following to /etc/httpd.conf:

```
/hod_publish_dir_alias/CustomizedCAs.p12
/user_publish_dir/CustomizedCAs.p12
```

10. Restart the Web server.
11. From a Web browser, specify the URL: http://your_HOD_server/user_alias/your_html.html.

Migrating on server operating systems with an uninstall program

On server platforms that have an uninstall program (for example, Windows and AIX), the uninstall program assists in the upgrade process. The uninstall program does not uninstall any files that the installation program did not install initially; such as, CustomizedCAs.class, CustomizedCAs.p12, or customized HTML files. Also, there are no changes to the private directory during the uninstall of the previous release. Any customized files that you added for the previous release of Host On-Demand remain unchanged when you install the new version of Host On-Demand. Run the uninstall program to remove the old version and then install the new version of Host On-Demand.

Refer to "Installing on Windows, AIX, Linux, Solaris, and HP-UX" on page 64 for installation instructions.

Migrating on server operating systems without an uninstall program

On operating systems without an uninstall program, you should delete the Host On-Demand 7 installation directory. *Before* you delete the installation directory, however, copy the private directory, any files added to the publish directory (such as CustomizedCAs.class, CustomizedCAs.p12, or customized HTML files), and the HODData directory to a temporary location. After you install Host On-Demand 8, move these files and directories back to their original locations.

Moving a Host On-Demand server installation to a new server

If you install Host On-Demand in a test environment before deploying to your production environment, complete the following steps to migrate Host On-Demand from one server to another (or from one HFS to a different HFS in an OS/390 or z/OS environment). First, install Host On-Demand on the new server. Then copy the private directory, any files added to the publish directory, such as CustomizedCAs.class, CustomizedCAs.p12, or customized HTML files, and the HODData directory from the test environment to the new server environment.

If your current environment is not OS/390 or z/OS and you want to move to an OS/390 or z/OS environment, this migration requires some additional steps. You can copy the private directory and the CustomizedCAs.class and CustomizedCAs.p12 files over to the new server directly. However, you should use the DWUnzip utility to correctly install the customized HTML files and the HODData directory.

Migrating from CustomizedCAs.class to CustomizedCAs.p12

Starting with Host On-Demand 8, you can no longer create or update the CustomizedCAs.class file on Windows and AIX platforms. The Certificate Management utility (IKEYMAN) only allows you to create or update a newer version of this file called CustomizedCAs.p12. When you upgrade to Host On-Demand 8, the Host On-Demand installation automatically detects the CustomizedCAs.class file, creates the CustomizedCAs.p12 file, and places it in the

publish directory. Both the CustomizedCAs.class and CustomizedCAs.p12 files remain in your publish directory and are available to clients of different versions.

If you have a separate user publish directory and not the default publish directory, you need to run the migration tool manually. From your publish directory, use the following command to run the migration tool and migrate the CustomizedCAs.class into the CustomizedCAs.p12 file:

```
..\jre\bin\java -cp ..\lib\sm.zip com.ibm.eNetwork.HOD.convert.CVT2PKCS12
\user_directory_path\CustomizedCAs.class hod
```

The command appears in this document on two lines; however, you should type it all on one line.

Once you have migrated to the new CustomizedCAs.p12 file, you may need to make future updates. In order for these updates to appear in the CustomizedCAs.class file for older clients, you must run a reverse migration utility. For Windows platforms, this utility runs automatically each time you open and close the IKEYMAN tool. For AIX, you must manually run the utility from your publish directory using the following commands:

```
../jre/bin/java -cp ../lib/sm.zip com.ibm.eNetwork.HOD.convert.CVT2SSLIGHT
CustomizedCAs.p12 hod CustomizedCAs.class
```

Note that the second command appears in this document on two lines; however, you should type it all on one line.

Unlike the CustomizedCAs.class, the CustomizedCAs.p12 requires a password by definition to open the file using the Certificate Management utility (IKEYMAN). If you create the CustomizedCAs.p12 file, use hod as the default password. If the Host On-Demand installation creates the CustomizedCAs.p12 file after detecting CustomizedCAs.class in your publish directory, it automatically configures the CustomizedCAs.p12 file with the hod password.

Upgrading the Host On-Demand client

Download client users load the new Host On-Demand client code the first time they point their browsers to the download client HTML file after the Host On-Demand server has been updated to the new version of Host On-Demand. They will be able to use the new features of Host On-Demand right away.

The cached client code detects that there is a newer version available on the server. Depending on how you set the cached client upgrade controls, users could be delayed in upgrading to the newer version. They will not be able to take advantage of the new features until their client code gets upgraded, but they can continue to use the older cached client code until then.

Upgrading Host On-Demand 4.x cached clients to Host On-Demand 7 or later

If you upgrade from Host On-Demand 4.x to Host On-Demand 7 or later, your clients will no longer be able to communicate with the server without upgrading.

If you need to manage network demand while upgrading cached clients, you can gradually move all of your Host On-Demand 4.x cached clients to Host On-Demand 7 (or later) by setting up two servers. One would be a Host On-Demand 4.x server and the other would be a Host On-Demand 7 (or later)

server. Configure all clients to access the Host On-Demand 7 (or later) server, and then add the HTML parameter HODServer to HODCached.html, or any of your customized cached client HTML files that are on the Host On-Demand 7 (or later) server. There are two sets of applet parameters defined in the HTML. Add the HODServer parameter to the set defined by the array cHod_AppletParams. You can do all of this using the Deployment Wizard on the Additional Parameters window; however, if you want to manually modify the HTML, the format for the parameter is:

```
cHod_AppletParams[7] =<PARAM NAME=HODServer
VALUE=http://yourhostname/alias/HODCached.html>
```

where *yourhostname* and *alias* are your Host On-Demand 4.x server's hostname and alias, or Publish, directory. Make sure that the index of the new cHod_AppletParams array element is in the correct sequence with the existing array elements.

The *HODServer* parameter works with the UpgradePercent and UpgradeURL parameters to manage client upgrades. If the cached client won't be upgraded on this connection attempt, it is redirected automatically to the Host On-Demand 4.x server specified in the *HODServer* HTML parameter. If a cached client will be upgraded, the Host On-Demand 4.x cached client is removed and the Host On-Demand 7 (or later) cached client is installed. Once the client is upgraded to Host On-Demand 7 (or later), the HTML parameter is ignored and the client is no longer redirected to the Host On-Demand 4.x server. After you have gradually upgraded all your cached clients, you no longer need the Host On-Demand 4.x server.

Be aware of the following when you upgrade cached clients from Host On-Demand 4.x to Host On-Demand 7 (or later):

- Cached clients are upgraded in the foreground. The upgrade in background option is ignored.
- If you have customized Host On-Demand 4.x `HODCached.html` and have called it something different, like `OurHTML.html`, do the following:
  1. Copy the Host On-Demand 7 (or later) version of `HODCached.html` to the file `OurHTML.html`;
  2. Add the `HODServer` parameter to `OurHTML.html`. The `HODServer` parameter should specify `http://yourhostname/alias/OurHTML.html` as the Host On-Demand 4.x server.
- You can copy the new `HODCached.html`, which includes the `HODServer` parameter, to `AutoHODCached.html` and `AutoHODLaunch.html`, in case these pages are bookmarked by the clients. The `HODServer` parameter in `AutoHODCached.html` should specify the `AutoHODCached.html` page on the Host On-Demand 4.x server. The `HODServer` parameter in `AutoHODLaunch.html` should specify the `AutoHODLaunch.html` page on the Host On-Demand 4.x server.
- If you are using language-specific HTML files (such as `HODCached_es.html`, `AutoHODCached_es.html`, `AutoHODLaunch_es.html`, etc.) you can also add the `HODServer` to these pages.

Upgrading custom HTML files

When you upgrade to a new release of Host On-Demand, it is not necessary to edit your existing Deployment Wizard files. Those files will continue to work as they always have. However, if you wish to take advantage of new features available in the Host On-Demand Deployment Wizard, you must edit your existing custom HTML files using the new Deployment Wizard.

Follow these steps to edit an existing HTML file:

1. Start the Deployment Wizard.
2. On the Welcome window, select Edit an existing HTML file and select the HTML file that you want to edit.
3. Go through the Deployment Wizard, selecting the options you want.
4. On the last page of the Deployment Wizard, write the custom HTML file out under the same name. For example, if you edited myCustom3.html, then write the file out under the name myCustom3.html.
5. Deploy your updated custom HTML file to your Host On-Demand server, replacing the previous one.

If your users have Java 2-enabled browsers, and you have custom HTML files that you created or last edited in Host On-Demand 6.0, IBM strongly encourages you to edit the HTML files with the new Deployment Wizard to receive the improved support for Java 2 environments.

If you are using the Cached Client or Web Start client and want to use the upgrade controls, do not add any additional components to the Preload Options when you edit the HTML file after an upgrade. See Cached Client Settings in the online help.

Upgrading from Java 1 to Java 2 on the client

You can upgrade to Host On-Demand 8 and Java 2 on the client at the same time. However, upgrading to Host On-Demand 8, then deciding to upgrade to Java 2 at a later time requires an additional download of the Host On-Demand cached client. To avoid this, install Java 2 before upgrading to the new version of Host On-Demand.

For additional information on planning for Java 2 on the client, refer to Chapter 4, “Planning for Java 2 on the client”, on page 27.

For information on obtaining a Java 2 plug-in, refer to “Obtaining a Java 2 plug-in for your clients” on page 33.

Upgrading your HTML files to support the Java 2 client

Upgrading is the process of converting HTML files generated by an earlier version of Host On-Demand to a format that runs successfully on the Host On-Demand 8 client. Upgrading allows you to take advantage of the new features provided by the Host On-Demand 8 client.

The statements in the following sections apply to emulator clients only. Also, the statements in this section apply both to the emulator cached client and the emulator download client, unless the statement specifically mentions one or the other.

Migrating HTML files from Host On-Demand 7

You do not have to migrate HTML files from Host On-Demand 7.

Host On-Demand 7 has the same concept of client Java level as Host On-Demand 8. Consequently, whether the HTML file was created using the Deployment Wizard from Version 7 or from Version 8, the Host On-Demand 8 cached client runs the HTML file in the same way. For more information on client Java level, refer to "Host On-Demand Java level" on page 33.

Migrating HTML files from Host On-Demand 6

Host On-Demand 6 does not have the concept of client Java level and provides limited Java 2 support. As a result, you *must* migrate some types of HTML files created with Host On-Demand 6.

Migrating Java 1 HTML files from Host On-Demand 6: If you created HTML files with the Host On-Demand 6 Deployment Wizard that your users run on Java 1 browsers, and you want to continue running these HTML files on Java 1 browsers, then you do not have to migrate the HTML files. You can use the HTML files as they are.

However, if you want to run these HTML files on Java 2–enabled browsers, then you must migrate the HTML files. To migrate these files, edit them with the Host On-Demand 8 Deployment Wizard and choose a Host On-Demand Java level of Java 2 or Auto Detect.

Migrating Java 2 HTML files from Host On-Demand 6: If you created HTML files with the Host On-Demand 6 Deployment Wizard that your users run with Java 2-enabled browsers, these files allow your users to run Java 2-enabled browsers. To do this, these files have downloaded and run a Java 1 version of the Host On-Demand client.

IBM recommends that you migrate these HTML files in order to take advantage of the advanced features available in the Java 2 version of Host On-Demand.

To migrate these files, edit them with the Host On-Demand 8 Deployment Wizard and choose a Host On-Demand Java level of Java 2 or Auto Detect.

Chapter 9. Uninstalling the Host On-Demand server

To remove Host On-Demand 8, follow the steps for your operating system.

z/OS or OS/390

Follow the instructions in the Program Directory for uninstalling the Host On-Demand server on zSeries servers.

OS/400

1. Sign on to OS/400 with the `QSECOFR` user profile or a profile with equivalent security authorities.
2. Enter `STRQSH` from the command line to start the Qshell interpreter.
3. Enter `cd /QIBM/ProdData/hostondemand/install`.
4. Run the following shell script according to your desired installation mode:

   Console

   `uninst400.sh`

   Silent `uninst400.sh -silent`

   For other uninstallation options, refer to Appendix D, "Native platform launcher command line options", on page 183.

Windows, AIX, Linux, Solaris, and HP-UX

Run your operating system's uninstall utility, with path name *your_install_directory*`/uninst/`, where *your_install_directory* is the directory where you installed Host On-Demand:

- `hoduninstall.exe` for Windows
- `hoduninstall_aix.bin` for AIX
- `hoduninstall_linux390.bin` for Linux/390
- `hoduninstall_linuxppc.bin` for Linux partitions on pSeries and iSeries
- `hoduninstall_linux.bin` for all other Linux versions
- `hoduninstall_solaris.bin` for Solaris
- `hoduninstall_hpux11x.bin` for HP-UX

You can run the utility in console mode by using the `-console` command line option. Otherwise, follow the uninstall wizard's GUI.

OS/2 Stop the Host On-Demand Service Manager by pressing `Ctrl+C` in the OS/2 window in which you started it. Close the window. Make sure that you save the important Host On-Demand files before migration. Refer to "Migrating on server operating systems without an uninstall program" on page 74 for more information. Then, delete the Host On-Demand directories.

Novell NetWare

From the console, enter `java -exit` to stop the Java NLM. Make sure that you save the important Host On-Demand files before migration. Refer to "Migrating on server operating systems without an uninstall program" on page 74 for more information. Then, delete the Host On-Demand directories.

Part 3. Configuring Host On-Demand

Chapter 10. Configuring Host On-Demand emulator clients

After installing Host On-Demand, you need to create HTML files and configure Host On-Demand sessions for your users.

Creating Host On-Demand HTML files

The best way to create and set up your HTML files for Host On-Demand is to use the Deployment Wizard. The Deployment Wizard allows you to easily create custom HTML files that contain all of the Host On-Demand features tailored for your environment. The following is a list of some of the many features that can be configured using the Deployment Wizard:

- **Configuration models**. Configuration models define the high-level approach you wish to follow with regard to where you define your sessions and where any user preferences are kept. For more information about configuration models, refer to Chapter 3, "Planning for deployment", on page 23.
- **Preloads**. Host On-Demand runs as an applet or application and must download code to the users' machines. By default, the Host On-Demand client downloads all of the components, but you may reduce the download size by removing those components that are not needed.
- **Cached client, Web Start client, or Download client**. Cached clients retain the code the first time users access the HTML file, and store it on the users' machines. The Web Start client allows you to run Host On-Demand without a browser. Download clients download the necessary applet files each time users access the HTML files.
- **Web page appearance (custom HTML templates)**. You can easily set up a template that the Deployment Wizard will use to generate your HTML files. This feature makes it easy to add your own background, banners, etc.
- **Host On-Demand Java Level**. Clients running Java 2-enabled browsers will need somewhat different HTML files than those running Java 1-enabled browsers. In the Deployment Wizard, you can select Java 1, Java 2, or Auto Detect. For more information see "Host On-Demand Java level" on page 33.
- **Cached Client/Web Start options**. When running the cached client or Web Start client, the code must be upgraded when newer versions of the client are available. There are a number of Deployment Wizard options that allow you to control when the upgrades occur.
- **Location of the Host On-Demand install (code base)**. Usually, Deployment Wizard files are placed in the Host On-Demand server's publish directory. However, sometimes it may be useful to put these files in a location that is independent of the Host On-Demand server so that they can be granted different security controls or make Host On-Demand server upgrades easier, for example.
- **WebSphere Portal**. WebSphere Portal provides a framework for plugging content extensions known as portlets into a Web site. Portlets are applications that organize content from various sources and display it on a single HTML file in a browser window. The HTML files that are used to launch Host On-Demand sessions can be deployed as portlets, allowing users to access Host On-Demand through a portal interface.
- **Windows Domain logon**. If your users are logged on to a Windows domain, this option automatically logs users on to Host On-Demand using their

Windows user name. This option is available only when using the configuration server-based model in the Deployment Wizard.

- **Session Manager APIs**. The Host On-Demand Session Manager provides JavaScript APIs for managing host sessions and text-based interactions with host sessions. These APIs are intended to provide support for embedding host sessions into a Web page using JavaScript and can be enabled with the Deployment Wizard.

In addition to creating custom HTML files with the Deployment Wizard, another way to access Host On-Demand is to use one of a number of predefined HTML files that are installed with your server. These predefined HTML files are general-purpose HTML files, and they all support the configuration server-based model. Note that Database On-Demand is only available using these predefined HTML files.

Note: To use the Web Start client, you must use the Deployment Wizard. Predefined files for this client type are not provided.

Configuring Host On-Demand sessions

In addition to setting up your HTML files, you will need to define sessions for your users. If you are using the HTML-based model, then you configure your sessions in the Deployment Wizard at the same time that you create the HTML files. Otherwise, if you are using the configuration server-based model or the combined model, or using one of the predefined clients, you will need to create groups, users, and sessions in the configuration server using one of the administration clients.

There is a full range of options available to you when you are configuring your sessions, regardless of whether you need to use the Deployment Wizard or one of the administration clients:

- **Session properties**. All of the session properties can be configured, including connection information, security, etc. Each of the fields may be locked to prevent users from updating them.
- **Runtime options**. When configuring a session, you can launch the session and configure features such as session size and placement, colors, toolbar customization, and macros. You can configure runtime options in the Deployment Wizard and the Full administration client.
- **Disabling user functions**. You can disable almost any of the functions that users normally receive as part of their Host On-Demand session, such as bookmarking, creating or running macros, etc.

Using the Deployment Wizard

The Deployment Wizard runs on a Windows platform. To start the Deployment Wizard, select one of the following ways:

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

The Deployment Wizard Welcome window appears.

The Deployment Wizard guides you through configuration choices and provides comprehensive help for the features. When you have finished selecting features, the Deployment Wizard creates the HTML and supporting files for you. These files need to be placed on the Host On-Demand server in a directory known to your Web server; usually, this directory is your Host On-Demand server's publish directory.

Distributing the Deployment Wizard output to your Host On-Demand server

If your Host On-Demand server is on a Windows or iSeries platform, you may be able to write your Deployment Wizard HTML and configuration files directly to your Host On-Demand server's publish directory. On the final screen of the Deployment Wizard, you can select where to write the generated files. You may select any local or network drive accessible by the machine where your Deployment Wizard is running. In this case, you would direct the Deployment Wizard output to a publish directory on the Host On-Demand server and specify an output format of *HTML*. Assuming that you have already defined your sessions, the HTML page is then ready to be accessed by your users.

Otherwise, if your Deployment Wizard cannot directly write to your Host On-Demand server, then you should select to have the Deployment Wizard generate a zip file for the output format. The Deployment Wizard will then produce a single zip file containing all of the HTML and supporting files. You will need to move the zip file to the Host On-Demand server and use DWunzip to explode the zip file into the desired publish directory. Assuming that you have already defined your sessions, the HTML page is then ready to be accessed by your users.

Host On-Demand Java level

Host On-Demand Java level is a required setting (introduced in Host On-Demand 7 as Client Java Type) in the Additional Options of the Deployment Wizard that identifies the type of browser that a client should use to run the generated Host On-Demand HTML file. For more information, refer to Additional Options in the online help. The choices for Host On-Demand Java level are:

- Java 1

  Click this option if all your clients run Java 1 browsers.
- Java 2

  Click this option if all your clients run Java 2–enabled browsers.
- Auto Detect

  Click this option if some of your clients run Java 1 browsers and others run Java 2–enabled browsers, or if you are not sure which type of browser your clients run. For example, Auto Detect is appropriate if your users connect to your Host On-Demand server through the Internet, because you cannot control whether a user runs a Java 1 browser or a Java 2–enabled browser.

Effects of Host On-Demand Java level on the cached client

This section discusses the effects of Host On-Demand Java level on the emulator cached client. The discussion is limited to the emulator cached client because it is the most widely used client. Other clients function similarly.

Java detection

When the user starts a browser and connects to an HTML file on the Host On-Demand server, the browser launches the client startup code that it finds in the HTML file and in related files on the server. The client startup code, running in the browser on the workstation, detects information such as the following:

- The Host On-Demand Java level setting in the HTML file.
- The vendor and version of the browser on the client workstation that is running the HTML file.
- Whether or not a Java 2 plug-in is installed on the client workstation.

Based on all these circumstances, and guided especially by the Host On-Demand Java level setting and the browser type, the client startup code makes a decision about whether to launch the Java 1 client or the Java 2 client.

Host On-Demand Java level: Auto Detect

When the Host On-Demand Java level is Auto Detect, Host On-Demand runs the version of the emulator cached client that matches the browser's Java type (either Java 1 or Java 2-enabled).

More specifically, if your user launches the HTML file using a Java 1 browser then Host On-Demand installs (if not already installed) and runs the Java 1 version of the Host On-Demand client. If your user launches the HTML file using a Java 2-enabled browser, then Host On-Demand installs (if not already installed) and runs the Java 2 version of the Host On-Demand client.

The following table summarizes these outcomes:

Table 14. Actions Taken When Host On-Demand Java level is Auto Detect

Host On-Demand Java level	Browser type	Action taken
Auto Detect	Java 1: • Netscape 4.7x • Internet Explorer without the Java 2 plug-in	Launch the Java 1 version of the emulator cached client.
Auto Detect	Java 2-enabled: • Netscape Version 6 or 7 • Internet Explorer with the Java 2 plug-in	Launch the Java 2 version of the emulator cached client.

The following sections contain additional information about using the Host On-Demand Java level of Auto Detect.

Users with Java 1 browsers cannot use Java 2-only features. As with all the Host On-Demand Java level settings (Java 1, Java 2, Auto Detect), your users with Java 1 browsers run the Java 1 version of the Host On-Demand client. Consequently, these users cannot take advantage of the Java 2-only features of the Host On-Demand client, such as the accessibility features, Auto-IME/On-the-Spot Conversion, and Print Screen enhancements.

Slightly longer startup time. When the Host On-Demand Java level is Auto Detect, the client startup time is slightly longer (1–2 seconds) because of the time

required for detection. Therefore, if you know that all your users run one type of browser, either Java 1 or Java 2-enabled, then you should use a Host On-Demand Java level of Java 1 or Java 2 rather than Auto Detect.

Handling of Internet Explorer with Java 2 plug-in. When a Java 2 plug-in is installed, Host On-Demand considers Internet Explorer on a Windows client to be a Java 2-enabled browser, even if the user does not know that a Java 2 plug-in is installed. Therefore, as Table 14 on page 86 shows, Host On-Demand runs the Java 2 version of the Host On-Demand client in this situation. For more information, refer to "Microsoft Internet Explorer with a Java 2 plug-in" on page 32.

Host On-Demand Java level: Java 1

The effect of using a Host On-Demand Java level of Java 1 in an HTML file is that Host On-Demand does not allow a user to run the HTML file unless the user is running a Java 1 browser.

If your user runs a Java 1 browser, then Host On-Demand installs (if not already installed) and runs the Java 1 version of the emulator cached client.

However, if your user runs a Java 2-enabled browser such as Netscape Version 6 or 7, and the emulator client is a cached client, then Host On-Demand displays the following error window.

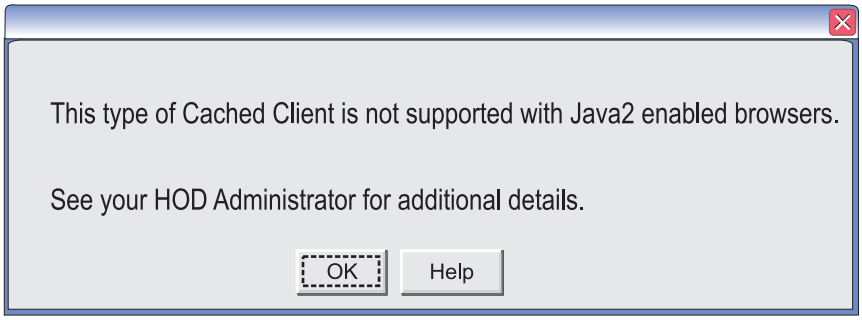

Figure 8. Error window for Host On-Demand Java level of Java 1 and Java 2-only browser

On this window:

- If the user clicks OK, then the cached client terminates.
- If the user clicks Help, then a window appears with further information. Refer to Host On-Demand Java level is Java 1 in the online help.

Finally, if your user runs Internet Explorer on a Windows platform with a Java 2 plug-in installed, the situation is different from the situation in which the browser is Netscape 6 or 7. With a Java 2 plug-in installed, Internet Explorer can function either as a Java 1 browser or as a Java 2-enabled browser. In this situation, as Table 15 on page 88 shows, when the Host On-Demand Java level is Java 1, Host On-Demand installs (if not already installed) and runs the Java 1 version of the emulator cached client.

The following table summarizes these outcomes:

Table 15. Actions Taken When Host On-Demand Java level is Java 1

Host On-Demand Java level	Browser type	Action taken
Java 1	Java 1: • Netscape 4.7x • Internet Explorer without the Java 2 plug-in	Launch the Java 1 version of the emulator cached client.
Java 1	Java 2-enabled: • Netscape 6 • Netscape 7	Display error window shown in Figure 8 on page 87 and do not run the HTML file.
	Java 2-enabled: • Internet Explorer with the Java 2 plug-in	Launch the Java 1 version of the emulator cached client on Internet Explorer's Java 1 JVM.

The following section contains additional information about using a Host On-Demand Java level of Java 1.

Users with Java 2-enabled browsers are excluded. As Table 15 shows, if one of your users has access only to a Java 2-enabled browser other than Internet Explorer, then that user cannot run the HTML file. Host On-Demand displays the message shown in Figure 8 on page 87.

The following sections describe problems you might encounter and how to solve them.

HTML file does not run on Internet Explorer on Windows platform. If a user sees the error window shown in Figure 8 on page 87 and is running Internet Explorer on Windows, then check to see if a Java 2 plug-in is installed. If a Java 2 plug-in is installed then check to see if the user has the Java 2 plug-in set as the default JVM for Internet Explorer. For more information on this problem see "Internet Explorer's default JVM must be the Java 1 JVM" on page 32. The Help window that is called from the window shown in Figure 8 on page 87 tells the user about this problem and how to solve it. See Host On-Demand Java level is Java 1 in the online help.

Host On-Demand Java level: Java 2

The effect of using a Host On-Demand Java level of Java 2 in an HTML file is that Host On-Demand tries to help users migrate from running a Java 1 browser to running a Java 2–enabled browser.

- If your user is running Netscape 4.7x, then Host On-Demand displays the following informational window:

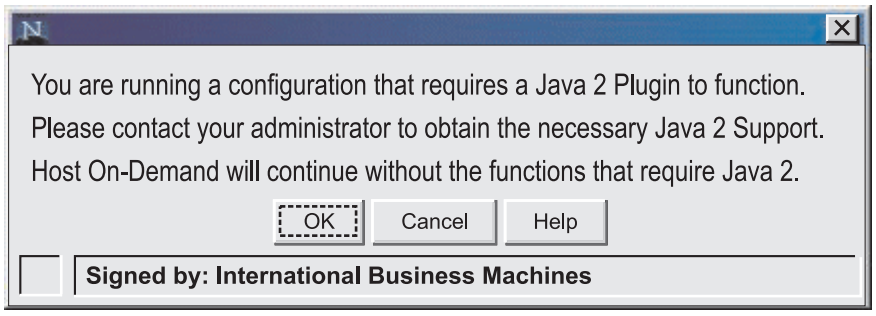

Figure 9. Error window for Host On-Demand Java level of Java 2 and Java 1-only browser (Netscape 4.7x)

On this window:

- If the user clicks OK, then Host On-Demand installs (if not already installed) and runs the Java 1 version of the emulator cached client.
- If the user clicks Help, a window appears with further information. For more information, refer to Host On-Demand Java level is Java 2 in the online help.

- If your user runs a Java 2-enabled browser, then Host On-Demand installs (if not already installed) and runs the Java 2 version of the emulator cached client.
- If your user runs Internet Explorer without the Java 2 plug-in installed on the Windows platform, then Host On-Demand displays the following informational window:

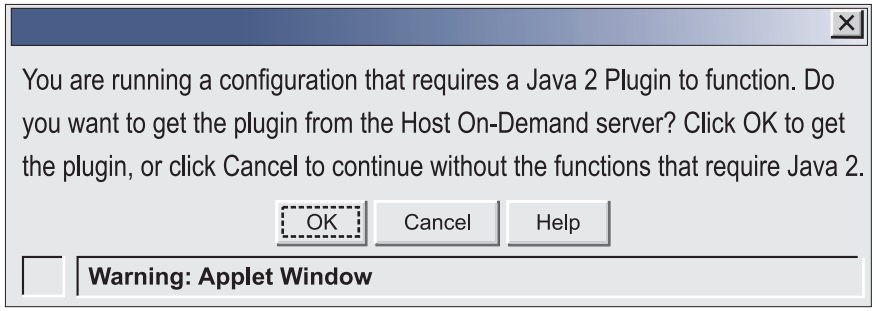

Figure 10. Error window for Host On-Demand Java level of Java 2 and Java 1-only browser (Internet Explorer)

On this window:

- If the user clicks Cancel, then Host On-Demand installs (if not already installed) and runs the Java 1 version of the emulator cached client.
- If the user clicks OK, then Host On-Demand displays a window from which the user can download the IBM Java 2 plug-in for the Windows platform. For more information, refer to "Obtaining a Java 2 plug-in for your clients" on page 33.
- If the user clicks Help, then an informational window appears with additional information, including detailed instructions for downloading and installing the IBM Java 2 plug-in. For more information, refer to Host On-Demand Java level is Java 2 in the online help.

In the Deployment Wizard, you can use the ForceJREInstall option to have Host On-Demand skip this informational window and immediately display the window that allows the user to download the IBM Java 2 plug-in for Windows. Go to the ForceJREInstall topic in the online help for more information.

The following table summarizes these outcomes:

Table 16. Actions Taken When Host On-Demand Java level is Java 2

Host On-Demand Java level	Browser type	Action taken
Java 2	Java 1: • Netscape 4.7x	• Display error window shown in Figure 9 on page 89 • If user clicks OK, run the HTML file using the Java 1 JVM.
	Java 1: • Internet Explorer on Windows without the Java 2 plug-in	• Display error window shown in Figure 10 on page 89 • If user clicks OK, go to an HTML that allows the user to download the IBM Java 2 plug-in for Windows. • If user clicks Cancel, run the HTML file using the Java 1 JVM.
Java 2	Java 2-enabled: • Netscape 6 • Netscape 7	Run the HTML file using the Java 2 JVM from the Java 2 plug-in.
	Java 2-enabled: • Internet Explorer on Windows with the Java 2 plug-in	Run the HTML file using the Java 2 JVM from the Java 2 plug-in.

Chapter 11. Using Host On-Demand administration and new user clients

Host On-Demand supplies several predefined clients for administering Host On-Demand and creating new user accounts. Before accessing an emulator client or a Database On-Demand client that uses the configuration server-based or combined deployment models, you must add users and configure sessions for them with one of the administration or full administration clients.

Loading administration and new user clients

To load an administration or new user client, do one of the following:

- Specify the full URL of the HTML file in your browser:

  `http://server_name/hod_alias/client_name.html`

  where *server_name* is the host name or IP address of the Host On-Demand server, *hod_alias* is the alias (or path) of the publish directory, and *client_name* is the HTML file name of the administration or new user client. For example, you can download the cached version of the administration client from the Web server by specifying a URL such as the following:

  `http://host.yourcompany.com/hod/HODAdminCached.html`

  To log on as the administrator the first time after the initial installation:

  1. Type the default user ID: admin.
  2. Type the default password: password.
  3. Click Log On.
- Load the HODMain_*xx*.html file, where *xx* is your two-letter language suffix, into your browser to view links to all the available administration and new user clients, plus other predefined clients. HODMain_xx.html is located in the publish directory.

Administration clients

Administration clients enable you to perform the following tasks for data stored on the configuration server:

- Manage users, groups, and sessions
- Configure, manage and trace the Redirector service
- Configure Database On-Demand
- Enable security
- View trace and message logs
- Disable functions to end users

Administration clients run on all Host On-Demand client platforms except the Macinstosh operating system. If you are creating HTML files in the Deployment Wizard using either the configuration server-based or combined models, you must configure sessions on the configuration server using an administration client. Refer to Basic Configuration Steps in the online help for more detailed information about configuring the Host On-Demand configuration server.

Host On-Demand supplies the following predefined administration and full administration clients:

There will be a delay using predefined HTML files if you use Internet Explorer only with Java 1. To avoid this delay, you can edit the HTML and change the hod_JavaType JavaScript variable from a value of 'detect' to 'java1'.

Administration client (HODAdmin.html)
Loads the download version of the administration client.

Administration client cached (HODAdminCached.html)
Loads the cached version of the Administration client. The advantage of using this client is that it can be cached along with the cached client in the browser.

To bookmark the cached Administration client, you must manually create the bookmark. It must point to HODAdminCached.html, so that Host On-Demand can compare the cached version to the server version. This allows Host On-Demand to recognize and notify you that a newer version of the cached Administration client is available at the server.

Administration client cached with problem determination (HODAdminCachedDebug.html)[1]
Loads the Administration client in a cached environment with problem determination (session logging and tracing) enabled.

Full Administration client (HODAdminFull.html)[2]
Loads the download version of the full Administration client. The full administration client gives the administrator the additional ability of starting sessions to configure runtime properties. However, the download size of the full administration client is larger than the download size of administration client.

Full Administration client cached (HODAdminCachedFull.html)[2]
Loads the cached version of the full Administration client. Like the cached version of the regular Administration client, this client can be cached along with the cached client in the browser.

Full administration client cached with problem determination (HODAdminCachedDebugFull.html)[1, 2]
Loads the cached version of the full Administration client with problem determination (session logging and tracing) enabled.

Notes:

1. Use the problem determination clients only if you are working with Support to resolve a problem with your Host On-Demand installation.
2. The full Administration client is the Administration client with Start Session enabled.
3. If you use a Java 2–enabled browser, you must use the Java Control Panel to remove the Administration cached client. For instructions, refer to Using the Java 2 plug-in in the online help.

Directory Utility

Directory Utility is a command-line Java application the administrator can use to manage user, group or session configuration information. This information is stored either in the Host On-Demand default data store, or in an LDAP directory. This utility is only useful in the environment where the Configuration Server-based model is in use. Directory Utility allows you to add, delete, or update large

numbers of users, groups, or sessions in a batch mode environment instead of using the Administration client. Directory Utility reads an XML ASCII file that contains the following actions to be performed on users, groups, or sessions defined to the Configuration Server:

- Add, update, and delete groups
- Add, update, and delete users from groups
- Add, update, and delete sessions from users or groups
- List existing users and groups in output files, as products of unique searches
- List existing users and groups in output files that can be reused as input

Searches performed with the list action are either user-based (returning user-specific information) or group-based (returning group-specific information). LDAP environments, however, support only user-based searches.

For more information, see Using the Directory Utility in the online help.

New user clients

If the administrator has enabled Allow users to create accounts in the Users/Groups window, users can use the predefined new user clients to create new accounts. See the New User client topic in the online help for more information about this client.

There will be a delay using predefined HTML files if you use Internet Explorer only with Java 1. To avoid this delay, you can edit the HTML and change the hod_JavaType JavaScript variable from a value of 'detect' to 'java1'.

The following new user clients are supplied with Host On-Demand:

New user client (NewUser.html)
Loads the download version of the New user client.

New user client cached (NewUserCached.html)
Loads the New User client in a cached environment.

New user client with problem determination (NewUserCachedDebug.html)[1]
Loads the New User client in a cached environment with problem determination (session logging and tracing).

Notes:

1. Use the problem determination clients only if you are working with IBM Support to resolve a problem with your Host On-Demand installation.

Chapter 12. Using Host On-Demand emulator clients

This chapter discusses issues that you need to be aware of when configuring and using Host On-Demand terminal emulator clients.

- "Loading emulator clients" describes how to access Host On-Demand emulator clients.
- "Selecting the appropriate client" on page 96 discusses how to decide which client is best for your needs.
- "Cached clients" on page 97 discusses how to use cached clients, including installing and removing them, comparing Java 1 and Java 2 cached clients, deploying them over the Internet, support for Windows 2000, Windows XP, and Mac OS X, and troubleshooting problems.
- "Web Start client" on page 107 discusses how to use the Web Start client, including installing and removing it, configuring your Web browser, using Web Start with Windows restricted users, and upgrading.
- "Download clients" on page 110 discusses how to use download clients, including installing them and loading them after downloading a cached client or Web Start client.
- "Predefined emulator clients" on page 111 describes the predefined emulator clients supplied with Host On-Demand.
- "Reducing client download size" on page 111 discusses strategies for reducing the download size of clients.
- "Deploying customer-supplied Java archives and classes" on page 112 describes how to deploy Java 2 archives and class files to your clients.

Loading emulator clients

To load a Host On-Demand emulator client, a user starts a Web browser and enters in the Address field the URL of a Host On-Demand HTML file. The Host On-Demand HTML file must be one of the following:

- An HTML file that you create with the Deployment Wizard.
- One of several generic predefined HTML files included with Host On-Demand

IBM recommends the first option. For more information on the Deployment Wizard, see the Deployment Wizard topic in the online help. For more information on the generic predefined HTML files, see "Predefined emulator clients" on page 111.

If your emulator client is deployed with the configuration server-based or combined deployment model, you must add users and configure sessions with the administration client before you can use the emulator client.

To launch HTML files generated by the Deployment Wizard, specify the full URL of the HTML file in your browser:

`http://server_name/hod_alias/client_name.html`

where *server_name* is the host name or IP address of the Host On-Demand server, *hod_alias* is the alias (or path) of the publish directory, and *client_name* is the HTML

file name of the client. For example, if you created an HTML file in the Deployment Wizard called 3270sessions.html, you can load it by specifying a URL such as the following:

```
http://host.yourcompany.com/hod/3270sessions.html
```

To launch a predefined HTML file included with Host On-Demand, point your browser to HODMain_*xx*.html file, where *xx* is your two-letter language suffix, to view links to all the available predefined clients. HODMain_xx.html is located in the publish directory.

When you access a client, a security warning appears to notify you that Host On-Demand was created by **International Business Machines**. Users must grant privileges in order for Host On-Demand to work properly.

Selecting the appropriate client

The types of Host On-Demand clients that you use depend on your computing environment and your personal preferences.

Cached clients and Web Start clients are stored locally and load faster than download clients (unless an updated version of the client is being downloaded from the Web server). You can use them equally well over network and dial-up connections. Cached clients and Web Start clients take up more local disk space than download clients, but on most machines this is not a problem.

The Web Start client allows users to run Host On-Demand sessions without a browser. Users start Host On-Demand sessions from the Java Web Start Application Manager. If a user closes the Host On-Demand desktop and there are active sessions running, the user is prompted to make sure he wants to close all sessions.

Download clients are generally used in LAN-connected environments because high-speed network connections reduce the time it takes to download them from the Web server. They are not recommended for use over low-speed dialup connections because they need to be downloaded every time they are used, which takes more time on dialup connections. The small disk footprint of download clients is especially well-suited for client machines that do not have a lot of local disk space, such as NetStation machines.

You can use cached, Web Start, and download clients in the same Host On-Demand environment, although you must remove Java 1 cached clients before you can load a download client. Refer to “Removing the cached client” on page 101 for instructions on removing cached clients.

If you plan to use the Web Start client, you must use the Deployment Wizard to generate your HTML file. If you plan to use cached clients or download clients, IBM recommends that you create your own clients using the Deployment Wizard instead of using one of the predefined clients. Refer to “Reducing client download size” on page 111 for more information.

Cached clients

A Host On-Demand cached client is any Host On-Demand client whose components have been cached (stored locally for quick access) on the hard disk of a user's workstation. When a user first runs a cached client, the Host On-Demand startup code downloads the Host On-Demand client components and stores them on the hard disk of the user's workstation. This is called installing the cached client.

When the user then runs the cached client, the Host On-Demand startup code downloads only a small startup applet from the server. The startup applet in turn starts the Host On-Demand client from the cached components on the hard disk.

By using the cached client, the user avoids having to wait for the Host On-Demand client components to be downloaded because they are already immediately available on the workstation's hard disk. In addition, the cached client is persistent across operating system restarts and browser reloads. Even though the cached client was originally intended for users with slow connectivity, such as dial-up phone lines, where downloading a large applet would take a long time, many customers have preferred using the cached client even for high-speed lines.

Like all Host On-Demand clients, the cached client is started (both the first time and subsequently) by specifying the URL of a Host On-Demand HTML file in the Address field of a supported Web browser. IBM recommends that you create your own HTML file using the Deployment Wizard. However, you can also use one of the generic, predefined cached client HTML files included with Host On-Demand.

The applet that starts the cached client also determines whether the version number of any of the Host On-Demand client components on the Host On-Demand server is newer than the version number of the corresponding downloaded components. If so, then the applet upgrades the cached client by downloading and caching the newer component from the server before launching the cached client.

The user can install multiple types of a cached client on the same workstation. For example, an emulator cached client, a Database On-Demand cached client, and an administration cached client could all be installed on one workstation. Also, with the Java 2 version of Host On-Demand (but not the Java 1 version), the user can install two versions of the same cached client: one with problem determination and one without problem determination.

Comparing Java 1 and Java 2 cached clients

If you are uncertain about the meaning of the terms Java 1 cached client and Java 2 cached client see "Terms relating to Java 1 and Java 2" on page x.

The Java 1 cached client and the Java 2 cached client have several key differences. For the Java 1 version:

- Only one version of the Host On-Demand Java 1 cached client can be installed. See "Java 1 cached client" on page 98.
- You cannot run a Java 1 download client while a Java 1 cached client is installed. To run a download client, you must first remove the cached client. For instructions on removing the cached client, see "Removing the cached client" on page 101.

In contrast, for the Java 2 cached client:

- Multiple versions of the Java 2 cached client can exist on the user's workstation, because the Java 2 cached client startup code installs a separate copy of the client onto the workstation's hard disk for each Host On-Demand server that the user visits.
- The Java 2 download client can be run without the user having to remove the Java 2 cached client.

In addition, improvements to the Java 2 version of the cached client have removed most of the previous limitations, such as the inability to download the cached client in the background. For more information, refer to "Improvements to the cached client for Java 2" on page 28.

Installing cached clients

You can install a cached client either from a Host On-Demand server or from a LAN drive or CD drive. These two methods work for both the Java 1 and Java 2 cached clients.

Information installed for the cached client

Two types of information are stored on the user's workstation when a Java 1 or Java 2 cached client is installed:

- Host On-Demand components

  These components are in the form of Java archive files, which are .JAR or .CAB files for a Java 1 cached client or .JAR files for a Java 2 cached client.
- Control information

  This information includes data such as the URL of the Host On-Demand server and the version of each downloaded component.

Java 1 cached client: For the Java 1 version of the cached client, only one version of the Host On-Demand cached client can be installed. However, the currently installed version of the Java 1 cached client can be updated if the user visits a Host On-Demand server that contains a newer version of the cached client. As a result, difficulties might arise when a user first installs the cached client from one server, such as ServerA, and later tries to access a different server, ServerB, that contains an older version of Host On-Demand. See "Cached client support issues when accessing multiple Host On-Demand servers" on page 103.

For the Java 1 version of the cached client, all types of the cached client that the user can install, such as emulator client, Database On-Demand client, and administration client, are installed in the same directory on the workstation's hard disk. This mixture of types of the cached client is natural because the different types share many components and differ only in a few key components.

Java 2 cached client: Multiple versions of the Java 2 cached client can exist on the user's workstation because the Java 2 cached client startup code stores the cached client components in a different directory of the workstation's hard disk for each server from which the user has downloaded a cached client. For more information, see "Comparing Java 1 and Java 2 cached clients" on page 97.

For the Java 2 cached client, all the client components that are downloaded from the same server are stored in the same directory on the user's hard disk. For example, if the user installs a Java 2 emulator client and a Java 2 Database On-Demand client from the same server, then the component files for both types of client are stored in the same directory. As with the Java 1 cached client, this mixture of types of the cached client is natural because the different types share many components and differ only in a few key components.

For a few specialized types of Java 2 cached clients, the client components are stored in the Java 2 plug-in's *sticky cache*. These are the same cached client types that are listed in "Improvement support limitations" on page 28.

Installing the cached client from the Host On-Demand Server

To install the cached client from a Host On-Demand server:

1. Specify the full URL of the HTML file in your browser, as described in "Loading emulator clients" on page 95.
2. If you want to use a predefined client, click on the cached client link after loading http://*server_name*/*hod_alias*/HODMain.html, where *server_name* is the host name or IP address of the Host On-Demand server and *hod_alias* is the alias (or path) of the publish directory.
3. The cached client begins installing immediately. A window shows the progress of the installation. The upper progress bar of this window shows the status of individual files as they download, while the lower progress bar shows the status of the overall installation.

The installation progress window does not appear for a few types of Java 2 cached clients. These are the same Java 2 cached clients that are listed in "Improvement support limitations" on page 28.

4. When the installation completes:
   - For the Java 1 cached client:

     The installation code prompts the user to restart the browser. When the user restarts the browser and links to the same URL, the Java 1 cached client is launched.
   - For the Java 2 cached client:

     The installation code immediately launches the Java 2 cached client. The user does not have to restart the browser.

Installing the cached client from a LAN or CD

You can now have some or all of your users initially download the cached client from a LAN drive or a CD. To install the cached client, the user has to access the LAN drive or CD only once. After the installation, the user connects with a Host On-Demand server in the usual way.

The advantages of this method are that the cached client components are installed on the user's workstation more quickly than they would be if they had to be downloaded from the Host On-Demand server, and that the user is not placing an additional load on the Host On-Demand server by downloading an entire set of cached client components.

This method is supported on most client platforms, including Java 1 cached clients. However, several Java 2 cached clients do not support this feature. The Java 2 cached clients that do not support this feature are listed in "Improvement support limitations" on page 28.

Limitations: The HTML file cannot specify a separate user publish directory. (If you specified a Code Base in the Deployment Wizard, the HTML file cannot be used to install the cached client from a LAN or CD drive.) Refer to the online help for more information about the separate user publish directory.

Steps for the administrator to create the CD or LAN image:

1. Use the File Name and Output Format window in the Deployment Wizard to create your customized *.html files (for example, MyHOD.html). If you need to

distribute the Deployment Wizard files to another server, you might want to select Output Zip to allow you to use DWunzip. For more information, see Using DWunzip in the online help.

2. For the Java 2 cached client, you can avoid having the user type in the hostname of the Host On-Demand server during installation by specifying the additional HTML parameter WebServerHostname in the Deployment Wizard. This HTML parameter is not needed for the Java 1 cached client. For more information see HTML parameters in the online help.
3. After loading the new Deployment Wizard files to your server, test the new files to make sure they function as expected.
4. Copy or FTP the following files from the publish directory of your Host On-Demand server installation to a network drive or CD (make sure you put the same version of Host On-Demand on the CD or LAN drive that you have on your Host On-Demand server):
   - MyHOD.html
   - MyHOD.jnlp (if it exists)
   - z_MyHOD.html (if it exists)
   - hoddetect*.html
   - hodlogo.gif
   - hodbkgnd.gif
   - Installer.html
   - Installer2.html
   - *.jar
   - *.cab
   - *.properties
   - *.js
5. Copy the following files and directories while preserving the directory structure:
   - msgs\cached_*.properties
   - HODData\MyHOD*.*

If you are using a CD for cached client installation, the CD must be distributed with the same guidelines as the License Agreement and Export and Import regulations because it contains encryption technology.

Steps for the user: After the administrator has set up the LAN drive or CD, the user must perform the following steps to install the cached client.

1. Prepare the client machine for installation by doing the following:
   - Get access to the LAN drive or CD drive.
   - Get the name and location of the HTML file, such as `f:\myPath\MyHOD.html`, that the system administrator has placed on the LAN drive or CD. (The HTML file has the same name and the same contents for all users. It is not specific to one user.)
   - *For the Java 2 cached client only,* find the hostname of the Host On-Demand server to which the user will attach after installing the cached client. For example, if the user will attach to `http://myHODServer/hod/MyHOD.html`, then the hostname is `myHODServer`. This information is not needed for the Java 1 cached client.

For the Java 2 cached client, the system administrator can eliminate this step by adding the HTML parameter WebServerHostname to the HTML file. See HTML parameters in the online help.

2. Run the HTML file:

   Type the path and name of the HTML file in the browser's address input field, such as:

   ```
   f:/mypath/MyHOD.html
   ```

3. *For the Java 2 cached client only,* when prompted by the installation code, enter the host name of the Host On-Demand server to which the user connects after installing the cached client. For example, if the user launches `http://myHODServer/hod/MyHOD.html`, then the hostname is `myHODServer`. This step is not needed for the Java 1 cached client.

For the Java 2 cached client, the system administrator can eliminate this step by adding the HTML parameter WebServerHostname to the HTML file. See HTML parameters in the online help.

4. Wait while the Host On-Demand cached client is installed from the LAN drive or the CD.
5. When prompted, restart the browser and point it to the HTML file of the same name on the Host On-Demand server, such as:

   ```
   http://myServer/hod/MyHOD.html
   ```

   The name of the HTML on the Host On-Demand server is the same as the name of the HTML file on the LAN or CD.

After completing these steps, the Host On-Demand cached client starts in the usual way.

Removing the cached client

The two methods available for removing the cached client are discussed in the following sections. The first is a method for removing Java 1 cached clients in particular; the second is general-purpose removal method.

Before you begin

Removing the cached client means erasing the information that was stored on the user's hard disk when the Java 1 or Java 2 cached client was installed.

A user running the Java 1 cached client can have only one version of the cached client on the workstation. In contrast, a user running the Java 2 version of the cached client has a separate version of the cached client for each Host On-Demand server for which he downloaded a cached client. For more information, refer to "Information installed for the cached client" on page 98.

Consequently, removing the Java 1 cached client removes the single existing version of the Java 1 cached client from the workstation. In contrast, removing the Java 2 cached client removes only the version of the Java 2 cached client that was downloaded from the server that the user visits when he does the removal. For example, if the user visits the server `http://myHODServerA/hod` to remove the Java 2 cached client on the user's workstation, then only the Java 2 cached client that was downloaded from `myHODServerA` is removed.

Finally, for both the Java 1 and the Java 2 cached client, removing the cached client removes all the types of cached client (such as emulation, Database On-Demand, and administration) associated with that installation.

For example, removing the Java 1 cached client from a workstation removes the emulation cached client, Database On-Demand cached client, and administration cached client from that workstation, if they are installed.

Similarly, removing the Java 2 cached client from a workstation while attaching to server `myHODServerA` removes the emulation cached client, Database On-Demand cached client, and administration cached client that were previously downloaded from server `myHODServerA`. However, only the cached client components downloaded from that server are removed. Cached client components from other servers, if any, are not removed until the user connects to that server and performs a remove.

Removing Java 1 cached clients

To remove any Java 1 cached client, follow these steps:

1. Start your browser.
2. Connect to HODMain.html on the Host On-Demand server. For example, connect to the following URL:

   ```
   http://myServer/HOD/HODMain.html
   ```
3. Click the following entry under Utilities:

   ```
   Remove Cached Client (Removes Java 1 only)
   ```

In addition, if *all* of the following circumstances apply then you *must* use this method, rather than the general-purpose method, to successfully remove the Java 1 cached client:

- You are running Internet Explorer.
- The Java 2 plug-in is installed on the workstation. (It might have been installed without the user's knowledge by some downloaded application that requires Java 2.)
- You want to remove the Java 1 cached client.

This method is required in these circumstances because Host On-Demand detects Internet Explorer as a Java 2-enabled browser and tries to remove the Java 2 cached client, instead of removing only the intended Java 1 cached client.

Removing Java 1 and Java 2 cached clients

The general-purpose removal method removes both the Java 1 cached client (except in the special case with Internet Explorer described in "Removing Java 1 cached clients") and the Java 2 cached client. Follow these steps:

1. Start your browser.

   Start a Java 1 browser to remove a Java 1 cached client, or start a Java 2-enabled browser to remove a Java 2 cached client.
2. Connect to HODMain.html on the Host On-Demand server. For example, connect to the following URL:

   ```
   http://myServer/HOD/HODMain.html
   ```

If you are removing a Java 2 cached client, you must connect to the same server from which you installed the Java 2 cached client to successfully remove it. For more information, refer to "Before you begin" on page 101.

3. Click the following entry under Utilities:

```
Remove Cached Client (If Java 2 detected, removes Java 2, else removes Java 1)
```

There is also an alternate and more direct way of performing this general-purpose removal. Follow these steps:

1. Start your browser.
2. Connect to HODRemove.html on the Host On-Demand server. For example, connect to the following URL:

   ```
   http://myServer/HOD/HODRemove.html
   ```

   This removes the cached client.

If you are removing a Java 2 cached client, you must connect to the same server from which you installed the Java 2 cached client to successfully remove it. For more information, refer to "Before you begin" on page 101.

Whichever general-purpose removal method you use, you will be prompted to clear the Java 2 plug-in's cache if you have removed the following Java 2 cached clients:

- Administration cached clients
- Cached clients on the Apple Mac OS X
- Emulator cached clients with JavaScript Session Manager API enabled (only Java 2 Netscape or Mozilla)

A window appears to notify you to clear the Java 2 plug-in's cache. For more information, refer to Using the Java 2 plug-in in the online help.

Removing a cached client shared by multiple users

If multiple users share a single cached client, and one of these users removes the cached client, then the cached client is removed for all users. For information on sharing a single cached client, refer to "Cached client support for Windows 2000 and Windows XP" on page 104.

Cached client support issues when accessing multiple Host On-Demand servers

The following sections detail issues and problems that might arise when cached client users access multiple Host On-Demand servers.

Java 1 cached client

Java 1 cached client users cannot download a component belonging to an older version of the Java 1 cached client: The problem arises in the following situation:

1. The Java 1 cached client was installed with a preload list. Therefore, only the client components named in the preload list were downloaded when the cached client was installed.
2. After installation, the user visits a Host On-Demand server running an older version of the Java 1 cached client than the version installed on the user's workstation.
3. Next, the user tries to use a function that requires a component that has not yet been downloaded.

In this situation, the Java 1 cached client will not download the required component because it belongs to an older version of the Java 1 cached client and might cause problems if combined with the components already downloaded from

the newer version. The Java 1 cached client refuses to download the required component and displays a message to the user explaining the problem.

There is no best course of action for proceeding. The user must remove the newer version of the cached client and install the older version.

This problem can easily arise in an environment where users access different servers across the Internet and the servers themselves (perhaps because the various servers are owned by different business partners) are running different versions of Host On-Demand. Host On-Demand Version 5.0.4 or later is required to run the cached client in this environment.

To avoid this problem, the system administrator can take some or all of the following actions:

- Select all the functions a user needs (across all sites the user accesses) in a preload list when you create an HTML file using the Deployment Wizard
- Use the disable function of the Deployment Wizard to disable all functions not in the preload list and the functions not needed by your users
- Create separate HTML files for different user groups
- Give your HTML files a name that identifies your company

Java 2 cached client

A Host On-Demand Java 2 cached client installs a separate copy of the cached client code for each Host On-Demand server that the user visits. Therefore there is no problem accessing servers at different service levels. With some versions of the plug-in, users may need to increase the size of their Java 2 cache if they are going to visit many Host On-Demand servers.

Java 1 and Java 2 cached clients

The following problems can occur with both the Java 1 and Java 2 cached clients.

Problem using locally stored preferences: If you are using locally stored preferences, the custom HTML files you create must have names unique to your company, because the HTML file names differentiate between the locally stored preferences of different sites. Using generic names could cause preference conflicts for your users.

See the Host On-Demand support Web site for more information: If you have problems managing cached client deployment on the Internet, go to http://www.ibm.com/software/webservers/hostondemand/support.html for more information.

Cached client support for Windows 2000 and Windows XP

On a multi-user Windows machine running either Windows 2000 or Windows XP operating systems and either of the following two browser/Java combinations listed below, users can download their own independent version of the cached client:

- Internet Explorer and the Microsoft JVM (Java 1)
- Any supported browser with a Java 2 plug-in

If the JavaScript API is enabled, the cached client cannot be shared for Netscape and Mozilla Java 2 browsers due to a technical limitation.

Alternatively, you can add the following parameters using the HTML parameters selection of the Advanced Options window of the Deployment Wizard:

- ShareCachedClient: allows users to share a single instance of the cached client
- SharedCachedDirectory: allows you to specify the directory location where the cached client is to be installed

When the cached client is shared but you do not specify a directory, the cached client is installed in the default directory \Documents and Settings\All Users\IBMHOD. If you specify a directory, for example SharedCachedDirectory=c:\ibm, the Host On-Demand cached client appends IBMHOD\HODCC to this string, and the cached client is installed in this new location, for example, c:\ibm\IBMHOD\HODCC. An administrator or power user must either create the install directory manually or perform the first install of the shared cached client. In either case, the administrator or power user must change the security settings for this directory so that restricted users have Read, Modify, and Write access. The Administrator can either change the security settings and then download the cached client to the directory, or download the shared cached client to the directory and then change the security settings. If the security settings are not updated and a restricted user attempts to install the shared cached client, the user receives an error message that indicates there may be a problem with the file system, and the restricted user will not be able to use or update the cached client.

Once the administrator or power user changes the security settings, a restricted user can log on to Windows and can either install the shared cached client or use (or update) a previously installed version of the shared cached client. Other restricted users can log on to Windows and use the cached client without having to download it from the Host On-Demand server again. They can also upgrade the shared cached client, if necessary. For Internet Explorer using the Microsoft JVM (Java 1), after the shared cached client is installed, any user that logs on to Windows to access the cached client for the very first time will need to restart the browser one extra time when prompted.

If you do not want restricted users to share the cached client, a separate instance of the cached client is downloaded to the user directory for each restricted user.

If an administrator or a power user downloads the previous version of the cached client, and you want to allow restricted users to access it, the administrator or a power user must use HODRemove.html to remove the previous version of the cached client, and then change the security settings to the shared cached client directory to Read, Modify, and Write for restricted users, as described above.

For information about removing a shared cached client, see “Removing a cached client shared by multiple users” on page 103.

Cached client support for Mac OS X (Java 2 clients only)

Cached clients have the following limitations on Mac OS X:

- Staging of Host On-Demand updates is managed on a per server basis.
- Preloading cached clients from a CD or LAN drive serves no function. When the browser is redirected to the real Web site, the plug-in considers that to be a distinct Web server and the client is cached again.
- Host On-Demand runs as an applet and must download code to the users' machines. The Host On-Demand client downloads all of the components, but you can reduce the download size by removing the components that you do not need. On Mac OS X, you cannot install additional components after the initial download.

- The Host On-Demand Java files used to run the Host On-Demand cached client on a Java 2-enabled Web browser are stored in the Java Runtime Environment (JRE) cache. To remove the cached client on Mac OS X, you must use the Java Control Panel to clear the JRE cache. For instructions, refer to Using the Java 2 plug-in in the online help.
- When running the cached client, the code must be upgraded when newer versions of the client are available. There are a number of Deployment Wizard options that allow you to control when the upgrades occur. These options are not available on Mac OS X.

The Java 2 cached client improvements do not apply to the Mac OS X Java 2 cached client. For more information, refer to "Improvement support limitations" on page 28.

Troubleshooting cached clients

If you find that you cannot load the cached client, follow the troubleshooting suggestions provided below.

Netscape 4.x

1. In the browser window, click Edit > Preferences > Advanced.
2. Check Enable Java.
3. Check Enable JavaScript.

Microsoft Internet Explorer 4.0.1

1. In the browser window, click View > Internet Options > Security.
2. Make sure that the Internet and Local Intranet zones are set to Medium security.

Microsoft Internet Explorer 5.5

After upgrading your browser from Microsoft Internet Explorer 4 to Microsoft Internet Explorer 5.5, you might receive security exceptions in the Java console. When you install the Cached Client, several files are stored into the browser's directory structure. When you upgrade Internet Explorer from Version 4 to Version 5, the browser will no longer know about the CAB files that contain the Host On-Demand cached code. Since the browser cannot find the CAB files, it tries to use the class files directly from the server, causing security exceptions. To resolve this issue, you should upgrade your browser, remove Host On-Demand using HODRemove.html, and then reinstall the product using HODCached.html.

Mozilla

With the Mozilla browser, if nothing happens when you try to install the cached client, or if the attempt to install the cached client fails, check the browser's settings. Make sure that Mozilla is not set to suppress popup windows that appear on top of or under the Navigator window. This setting prevents the Host On-Demand cached client from being installed.

This location of this setting depends on the version of Mozilla:

- In Mozilla 1.2, this setting is included under Edit > Preferences > Advanced > Scripts & Plugins.
- In Mozilla 1.3, this setting is included under Edit > Preferences > Privacy & Security > Popup Windows.

After the cached client is installed, you can restore this setting to suppress popup windows. But if you need to install the entire cached client again or update to a

newer version in the foreground, you must set Mozilla again so that it does not suppress popup windows.

The setting to suppress popup windows does not hinder the downloading of additional components that were not included in the initial download (preload list).

Web Start client

The Java Web Start client allows users to start Host On-Demand without a browser. You must use the Deployment Wizard to generate a HTML file for the Web Start client. The HTML file generated by the Deployment Wizard points to a Java Network Launch Protocol (JNLP) file. The JNLP file defines a Java Application, including parameters passed to the application and the archives that contains class files used by the application. The JNLP file and the associated archives are stored on a Web server.

When a user points to the JNLP file, the browser launches the Web Start application on the client computer. It downloads the associated archives, checks to insure that the minimum required JRE is present (if specified), stores the archives on the user's machine, sets up icons to represent the application, and launches the application.

Users can start Host On-Demand sessions from the Java Web Start Application Manager. By using the Java Web Start Application Manager, Host On-Demand sessions do not depend on a browser. Therefore, closing a browser does not end a Host On-Demand session. If the user attempts to close the Host On-Demand desktop and there are active sessions running, the user is prompted to make sure he wants to close all sessions. If so, the sessions are terminated cleanly to prevent problems that occur when there are sessions running in the browser and the browser is abruptly closed.

After the initial launch of the application, you can either point the Web browser at the JNLP file again, or click the mouse on the icons created on the client machine. After Web Start is restarted, it checks the Web server for updates to the archives and downloads any updated files.

Java Web Start is bundled with JRE 1.4.0 and 1.4.1. If you use JRE 1.3, then you should upgrade to JRE 1.4. For more information about Java Web Start, refer to http://www.javasoft.com.

The Host On-Demand Web Start client has the following requirements:

- JRE 1.4 or later is required to use HTTPS to access files from the Web server.
- JRE 1.4 or later is required to use an HTTP proxy with Web Start.
- Session properties that say use Browser settings (like proxy server or TLS/SSL) cannot be used with Web Start.

Installing the Web Start client

You can install the Web Start client either from a Host On-Demand server or from a LAN drive or CD drive. These two methods of installation are described in the following sections.

Installing the Web Start client from the Host On-Demand server

Installing the Web Start client from the Host On-Demand server requires a browser. To install the Web Start client from a Host On-Demand server, perform the following steps:

1. Specify the full URL of the HTML file in your browser, as described in "Loading emulator clients" on page 95.

   The Web Start client begins installing immediately. A window shows the progress of the installation. The upper progress bar of this window shows the status of individual files as they download, while the lower progress bar shows the status of the overall installation.
2. When the installation completes, the installation code immediately launches the Web Start client. The user does not have to restart the browser.

Installing the Web Start client from a LAN or CD

Some or all of your users can initially download the Web Start client from a LAN drive or CD. The user must access the LAN drive or CD only once to install the Web Start client. Afterwards, the user connects to the Host On-Demand server as usual.

When installing from a LAN or a CD, the Web Start components are installed on the user's workstation more quickly than if they were downloaded from the Host On-Demand server. In addition, that user is not placing an additional load on the Host On-Demand server by downloading an entire set of Web Start client components.

Host On-Demand supports installing the Web Start client from a LAN or CD on all client operating systems.

If you are an administrator and you want to set up the LAN or CD for users, refer to "Steps for the administrator to create the CD or LAN image" on page 99.

If you are a user who wants to install the Web Start client from a LAN drive or CD, refer to "Steps for the user" on page 100.

Installing the Web Start client without a browser

When you select Web Start as your client type on the Additional Options window in the Deployment Wizard, the Web Start Settings window launches. On the Web Start Options tab, you must specify a code base. The HTML file generated by the Deployment Wizard points to a Java Network Launch Protocol (JNLP) file. The JNLP file's code base points to the Host On-Demand server. If you distribute the JNLP file, for example, `myhod.jnlp`, to each client or a LAN drive, your users can launch this file to install the Web Start client without using a browser. This installation method requires a connection to the Host On-Demand server.

On Windows with Java Web Start installed, if the user types the command `start myhod.jlnp` , the Host On-Demand session defined in the JNLP file installs and runs. At a command prompt on Linux, a user can type `/javaws http://HODServer/HODAlias/myhod.jnlp` to install and run the Host On-Demand session. A Host On-Demand icon appears in the Java Web Start Application Manager. The user can double click this icon to launch Host On-Demand.

Configuring your Web server for Web Start

The administrator must register the JNLP extension as a mimetype with the Web server so the browser knows to launch the Web Start application. For example, the following sections describe how to configure Apache HTTP Server, IBM HTTP Server, and Microsoft IIS.

Apache HTTP Server or IBM HTTP Server

To configure the Apache HTTP Server or IBM HTTP Server for Web Start, add the following line to mime.types:

```
application/x-java-jnlp-file    JNLP
```

Microsoft IIS 5.x

To configure Microsoft IIS for Web Start, complete the following steps:

1. From Control Panel > Administrative Tools > Internet Information Services, click Default Web Site.
2. Click the HTTP Headers tab on the Properties.
3. Under MIME Map, click the File Types tab and select New Type.
4. In the Extension field, type `.jnlp`.
5. In the Content Type field, type `application/x-java-jnlp-file`.
6. Click OK.

Upgrading the Web Start client

After the initial install of the Web Start client, if users point their browsers to the HTML file generated by the Deployment Wizard and there are updates available on the Host On-Demand server, Host On-Demand prompts the user to update. If the user wants to update, Java Web Start downloads the updated archive files and launches Host On-Demand. If the user declines to upgrade, Host On-Demand prompts him again the next time he launches the HTML file.

Adding Web Start components after the initial install

If users request a function that is not installed on the Java Web Start client, Host On-Demand prompts them to install the additional components required for that function. If they choose to install the additional components, they must restart Host On-Demand to use them.

Web Start and Windows Restricted Users

Windows Restricted Users with Java Web Start 1.0.1 should remove the JRE and Java Web Start and reinstall a newer JRE with Java Web Start 1.2.

Bookmarking sessions with Web Start

Since the Web Start client runs outside of a browser, bookmarking is disabled since bookmarking is a browser feature. Administrators can create Web Start clients that give users the same look as running an embedded bookmarked session by doing the following:

1. On the Advanced Options window of the Deployment Wizard, add the HideHODDesktop parameter with a value of true.
2. Configure a single session to autostart.
3. Configure the session to not start in a separate window.

Using Web Start with HTTPS

If you want to use HTTPS with the Web Start client, the certificate authority used for your secure HTTP connection should come from a well known root authority. When you use Host On-Demand as an applet and use an HTTPS connection, you are given the opportunity to trust the certificate used for the HTTPS connection if the root authority is not known by the browser. Since Java Web Start runs as an application, this browser facility is not available. The Java Virtual Machine used by Java Web Start contains several root authorities that it trusts. If the certificate that comes from the HTTPS connection has a root authority of one of these authorities known by the JVM, the secure connection can be established. If you want to use a certificate authority other than ones known by the JVM by default, for example, a self-signed certificate, you must import the certificate into the keystore of the JVM for each of the clients accessing this Java Web Start client. This is required to establish the secure HTTP connection.

Removing the Web Start client

To remove the Web Start client, complete both of the following steps:

1. In the Java Web Start Application Manager, highlight your application and click Remove.
2. Launch HODRemove.html in your browser.

Download clients

Unlike the cached client and Web Start client, the download client does not control how or when client components are downloaded to the workstation's hard disk. The download client leaves all caching decisions to the browser.

Use the download client if you meet *both* of the following requirements:

- You do not want to take up disk space on client machines by installing the cached client or Web Start client.
- Your initial download time is not an issue.

Launching the download client

Launch the download client by downloading it from the Host On-Demand server into your browser window, as described in "Loading emulator clients" on page 95.

Launching the download client after installing the cached client or Web Start client

Java 1

If you have installed a cached client and then later decide to launch a download client, you must first do the following:

1. Remove the cached client from the browser by loading HODRemove.html in your browser, as described in "Removing the cached client" on page 101.
2. Restart your browser.

If you do not remove the cached client before loading the download client, the session will not start and an error message appears directing you to run HODRemove.html before you can launch the download client.

Java 2

With Java 2 clients, you can successfully launch the download client after installing the cached client or Web Start client.

Predefined emulator clients

Several predefined emulator client HTML files are supplied with Host On-Demand. They are included to demonstrate the range of Host On-Demand client functionality and to serve as examples for creating customized HTML files in the Deployment Wizard. All of them use the Configuration server-based model. To load one of these clients, follow the instructions in "Loading emulator clients" on page 95.

In general, it is recommended that you define your own customized HTML files with the Deployment Wizard instead of using the predefined client HTML files.

The following predefined emulator client HTML files are provided by Host On-Demand:

There is a delay using predefined HTML files if you use Internet Explorer only with Java 1. To avoid this delay, you can edit the HTML and change the hod_JavaType JavaScript variable from a value of 'detect' to 'java1'.

Cached client (HODCached.html)
Provides all Host On-Demand client functions.

Cached client with problem determination (HODCachedDebug.html)[1]
Starts the cached client with problem determination (session logging and tracing).

Download client (HOD.html)
Provides all Host On-Demand client functions except problem determination.

With a Java 2–enabled browser the predefined download client file HOD.html omits some infrequently used Host On-Demand components. For more information, including a list of excluded components and a description of workarounds, see "HTML files do not contain some components" on page 29. Accessing HOD.html with a Java 2 browser works with limited functions.

Download client with problem determination (HODDebug.html)[1]
Loads the download client with problem determination (session logging and tracing).

Notes:

1. Use the problem determination clients only if you are working with IBM Support to resolve a problem with your Host On-Demand installation.

Reducing client download size

In general, it is a good idea to keep the size of your Host On-Demand clients (whether download, Web Start, or cached clients) as small as possible. This speeds up their download time and conserves disk space on client machines.

The best way to minimize the size of your Host On-Demand clients is to create them by using the Deployment Wizard. The predefined clients supplied with Host On-Demand are typically larger than the custom clients created with the Deployment Wizard because they contain Host On-Demand's full range of client functionality. Clients created in the Deployment Wizard contain only the functions that you select to be pre-installed. In addition, Deployment Wizard clients are downloaded in compressed format. This further reduces their download size.

When you create a customized client with the Deployment Wizard, you can select only the functions that you know users are going to need on the Preload Options window in the Deployment Wizard. For instance, if your users are only going to need 3270 terminal and 3270 printer sessions, do not select any other session types when you are creating the client in the Deployment Wizard. Including support for unused session types increases the size of the client without improving its functionality.

If you click Auto Select on the Preload Options window, the Deployment Wizard selects the components you need based on your session configuration.

You can also choose not to download components for functions that are not frequently used. Unless you choose to disable that function in the Deployment Wizard, users will be prompted to download any necessary components when they use that function. If you need additional session types later, you don't necessarily have to create a new client type. You can add the new session types to the preload list on the Preload Options window instead.

On Mac OS X, you cannot install additional components after the initial download. For more information, refer to "Cached client support for Mac OS X (Java 2 clients only)" on page 105.

Do not use debugging or problem determination in either Deployment Wizard-generated or predefined clients. This greatly increases the size of the client and can slow down a client's performance. Debugging and problem determination clients are not intended for general use. Use them only in conjunction with Host On-Demand technical support to diagnose and solve problems with your Host On-Demand system.

Deploying customer-supplied Java archives and classes

Customer-supplied Java classes and archives are Java class files and archive files that are not included either as part of the Host On-Demand client or as part of the Java 1 or Java 2 Runtime Environment. Examples of such files are Java classes or archives that you yourself have implemented or that you have obtained from third parties.

You would want to deploy such classes or archives for use with the emulator client in the following situations:

- You want your users to run macros that call customer-supplied Java methods.
- You want your users to run a customer-supplied applet with the session (either started automatically with the session or launched using the Actions > Run Applet... selection on the menu of the session window).

For Java 2 limitations on running customer-supplied applets, see "Limitations with customer-supplied applets and Java 2" on page 30.

Although several methods are available for deploying these files, each method works only under certain circumstances. The possible methods are:

- Using the AdditionalArchives HTML parameter in the Deployment Wizard. See "Using the AdditionalArchives HTML parameter" on page 113.
- Copying the files to the Host On-Demand server's publish directory. See "Publish directory" on page 114.

- Putting the files into a directory that exists in the CLASSPATH environment variable (Java 1 only, Windows client platform only). See "Classpath" on page 114.

The deployment method you choose depends on:

- The type of file deployed (Java 1 classes, Java 1 archives, Java 2 classes, Java 2 archives)
- Where the files will be deployed (Host On-Demand server or client workstation)
- The type of client platform and the type of browser.

The following table shows which methods are available for each set of circumstances. An entry of (None) means that no method is available for that set of circumstances.

Table 17. Methods for deploying customer-supplied Java archives and classes

Server and clients	Java 1 class files	Java 1 archives (.CAB or .JAR)	Java 2 class files	Java 2 archives (.JAR)
• Cached client • Files on server	(None)	AdditionalArchives HTML parameter	(None)	AdditionalArchives HTML parameter
• Cached client • Files on client	(None)	Classpath, Java 1 only, Windows only, Netscape 4.x only.	(None)	(None)
• Download client • Files on server	Publish directory	AdditionalArchives HTML parameter	(None)	AdditionalArchives HTML parameter
• Download client • Files on client	(None)	(None)	(None)	(None)

The three methods available for deploying customer-supplied Java archives and classes are described in the following sections. In addition, "Hints and tips for archive files" on page 114 provides more information about using archive files.

Using the AdditionalArchives HTML parameter

You can use this method when you want to deploy Java 1 or Java 2 archives to a Host On-Demand server. This method works for the cached emulator client, the download emulator client, and for the Web Start client.

Java 1 archives must be either .CAB files (for Internet Explorer) or .JAR files (for Netscape and Mozilla). Java 2 archives must be Java 2 .JAR files.

The advantage of using the AdditionalArchives HTML parameter is that it causes your Java archives to be downloaded to the user's workstation automatically when one of your users connects with the cached client or download client HTML file on your Host On-Demand server.

The disadvantage of this method is that these Java archives or class files will be downloaded again every time your user connects to that HTML file. This means that even if your users are running the cached emulator client, these Java archives or class files will be downloaded every time your user connects to the HTML file. The reason for downloading the archives every time your user connects is to ensure that the Host On-Demand client has the latest versions of your archives or class files. As a result, this method works best when the Java archives or class files are relatively few and relatively small, so that your users do not have to wait a

long time for these files to be downloaded, and so that downloading these files to your users does not place a heavy load on your Web server.

To use this method, perform the following steps:

1. Place the archives in your Host On-Demand publish directory. The default publish directory is the subdirectory HOD in your Host On-Demand server's install directory, such as `c:\Program Files\IBM\HostOnDemand\HOD\`.
2. Edit the HTML file with the Deployment Wizard. Then:
   a. On the Advanced Options panel, click HTML Parameters.
   b. In the Name field, enter `AdditionalArchives`.
   c. In the Values field, enter the names of your Java archives, separated by commas, without file extensions (.cab or .jar). For example:

      ```
      myCustomA,myCustomB,MyCustomC
      ```

For more information, see AdditionalArchives in the online help.

Publish directory

This method works in the following situations:

- When you want to deploy Java 1 class files to a Host On-Demand server. However, this method works only for the download emulator client, not for the cached client.
- When you want to deploy Java 2 class files to a Host On-Demand server. The Java 2 class files must not belong to any Host On-Demand package.

You can use this method when you want to deploy Java 1 class files to a Host On-Demand server.

To use this method, place the archives in your Host On-Demand publish directory. The default publish directory is the subdirectory HOD in your Host On-Demand server's install directory, such as `c:\Program Files\IBM\HostOnDemand\HOD\`.

Classpath

You can use this method when you want to deploy Java 1 .JAR files to the client workstation. However, this method works only for Netscape 4.x running on a Windows client platform.

To use this method, perform the following steps for *each* client workstation on which you want the archives to be available:

1. Find the directory where Netscape 4.x is installed. For example:

   ```
   c:\Program Files\Netscape
   ```

2. Under this directory, find the `Program\java\download` directory where downloaded .JAR files are kept. If the download client has been run for this workstation using Netscape 4.x and Netscape 4.x's caches have not been cleared, then this directory will already contain the Host On-Demand emulator client archives, such as hodimg.jar. For example,

   ```
   c:\Program Files\Netscape\Communicator\Program\java\download
   ```

3. Copy your Java 1 .JAR files to this directory.

Hints and tips for archive files

The following hints and tips might provide helpful information about using archive files:

- When you create your archive (.jar or .cab), verify that the path of each class file is correct. For example, the path for com.mycompany.MyClass should be `com\mycompany\`. It should *not* be `C:\MyTestDirectory\com\mycompany\`, and it should not be blank (since the class file is part of a package).
- Verify that the proper permissions are set for your archive files. That is, in operating systems that use file permissions, such as Linux, AIX, Unix, and z/OS, the file permissions for the archive files should be set to 755 (that is, rwxr-xr-x).
- If you have two different cached client pages that specify different AdditionalArchives parameters, you must close and restart the browser when switching from one page to another. Otherwise, when you switch from one page to another, the cached client is not reloaded and, as a result, the AdditionalArchives parameter is not checked.

Chapter 13. Using Database On-Demand clients

Database On-Demand is a Java applet that allows users to perform Structured Query Language (SQL) requests to iSeries databases through a JDBC driver. Database On-Demand is shipped with a JDBC driver for the iSeries. Other user-installed JDBC drivers can be registered and used, although IBM does not provide support for these drivers.

Features of Database On-Demand include:

- A graphical interface to aid in constructing SQL statements and File Upload statements
- The ability to display on screen the results of the executable statements you build, to save the results of SQL statements in various file formats and to upload entire files in various formats to a host database
- The ability to create dynamic queries, using the graphical interface, that can be executed or saved for later use

You cannot create Database On-Demand clients using the Deployment Wizard. Database On-Demand clients are only available using the predefined clients.

For more Database On-Demand overview information, see Database On-Demand in the Host On-Demand online help.

Loading Database On-Demand clients

To load a Database On-Demand client, do one of the following:

- Specify the full URL of the HTML file in your browser:

  `http://server_name/hod_alias/client_name.html`

  where *server_name* is the host name or IP address of the Host On-Demand server, *hod_alias* is the alias (or path) of the publish directory, and *client_name* is the HTML file name of the Database On-Demand client. For example, you can load the download version of the Database On-Demand client from the Web server by specifying a URL such as the following:

  `http://host.yourcompany.com/hod/HODDatabase.html`
- Load the HODMain_*xx*.html file, where *xx* is your two letter language suffix, into your browser to view links to all the available Database On-Demand clients, plus other predefined clients. HODMain_xx.html is located in the publish directory.

Database On-Demand clients

Host On-Demand supplies the following predefined Database On-Demand clients:

There will be a delay using predefined HTML files if you use Internet Explorer only with Java 1. To avoid this delay, you can edit the HTML and change the hod_JavaType JavaScript variable from a value of 'detect' to 'java1'.

Database On-Demand client (HODDatabase.html)
Provides users with a means of making Structured Query Language (SQL)

requests to iSeries databases through a Java database connectivity (JDBC) driver. Users can save the results of their requests and use them in other applications, such as a spreadsheet.

Database On-Demand client cached (HODDatabaseCached.html)
This client starts the Database On-Demand client in a cached environment. The advantage of the Database On-Demand cached client is that it can be cached along with the Host On-Demand cached client in the browser.

If your client is going to use multiple code pages, you need to add the appropriate archive (.jar/.cab) file of each code page to the preload list of your cached HTML. For a list of code page languages and corresponding file names, see "Using multiple code pages with Database On-Demand".

Database On-Demand client cached with problem determination (HODDatabaseCachedDebug.html)
This client starts the Database On-Demand client in a cached environment with problem determination. Load this HTML file if you want to use the Database On-Demand client in cached environment with problem determination (session logging and tracing).

Use the problem determination client only if you are working with IBM Support to resolve a problem with your Host On-Demand installation.

Setting up Database On-Demand users

To configure users so they can access Database On-Demand, you must first define groups and users in the Host On-Demand configuration server. Then you can define the database functions that groups and users can perform and later manage the statements that users have created. The administrator cannot create SQL statements for users.

If you are using Database On-Demand with Netscape 4.x, you must turn the Just In Time (JIT) compiler off. Unfortunately, due to problems found with the JIT compiler, this means that you cannot take advantage of both the Database On-Demand and integrated Windows domain logon functions.

For more detailed information about setting up groups and users to access Database On-Demand, see the topics Getting started with Database On-Demand and Setting up options for Database On-Demand users in the Host On-Demand online help.

Using multiple code pages with Database On-Demand

If you wish to use multiple code pages with Database On-Demand, you must add jar or cab files to your HTML file. Only those code pages that correspond to the language of the HTML file are automatically loaded. For example, if you are running from a French computer, but you want to access a Dutch host, you must make these modifications.

Edit the CommonJars.js file. If you are using a download client, look for the line that starts "dbaDownloadJars =" and add the appropriate file names from the table below. Use jar file names, even if your clients will be using Internet Explorer (the names will be converted to cab file names later). If you are using a cached client, look for the line that starts "dbaCachedComps =" and add the appropriate component name from the table below.

Supported Database On-Demand code pages

The following table lists the supported Database On-Demand client code page languages, the corresponding .jar file names, and the cached component names:

Code page language	.JAR file name	Component name
Arabic	hacpar.jar	HACPAR
Czech, Hungarian, Polish, Slovenian	hacpce.jar	HACPCE
Danish, Finnish, Dutch, Norwegian, Swedish	hacp1b.jar	HACP1B
German, Spanish, French, Italian, Portuguese, Brazilian Portuguese	hacp1a.jar	HACP1A
Greek	hacpgr.jar	HACPGR
Hebrew	hacphe.jar	HACPHE
Japanese	hacpja.jar	HACPJA
Korean	hacpko.jar	HACPKO
Russian	hacpru.jar	HACPRU
Simplified Chinese	hacpzh.jar	HACPZH
Thai	hacpth.jar	HACPTH
Turkish	hacptr.jar	HACPTR
Traditional Chinese	hacptw.jar	HACPTW

Chapter 14. Creating and deploying server macro libraries

Server macro libraries are available for HTML model pages only. They allow you to create and maintain a central repository of macros for users to access from their Host On-Demand sessions. These macros are not downloaded to the user's machine until they are needed. When you make changes to a server macro, users automatically get your updates the next time they access the macro.

Server macro libraries have several benefits:

- They provide a convenient way to store, edit, and administer macros, all from one easy-to-access location.
- They allow easy sharing of macros among multiple users and across any number of sessions.
- They eliminate the need to import macros into the Host On-Demand session, and can therefore reduce the size of the session. The macros are only downloaded to the user's machine if and when the user accesses them.
- You can edit macros and replace the files in the server macro library at any time without regenerating Host On-Demand sessions or modifying the HTML files. Any changes you make are automatically available the next time a user requests that macro.

Server macro libraries are read-only for users and can reside on a Web server or on a shared network drive. For both types of libraries, you can control which macros are available to particular Host On-Demand sessions. If you use a Web-based macro library, you need to create a text file that identifies the specific macros that you want to be available for the session that you are configuring. If you use a shared drive-based macro library, then *all* the files in the specified directory will be available to the session. Which type of server macro library you choose depends on your computing environment as well as your company needs.

Deploying a server macro library to a Web server

1. Put your macros in a place that users can access through a Web server. This does not need to be the Host On-Demand publish directory.
2. For each session that requires a separate set of macros, create a text file that contains the list of the macro file names. The text file format can only have one macro file name per line, for example:

   ```
   macro1.mac
   macro2.mac
   macro3.mac
   ```

   Be sure to note the following rules:

   - The macro name must be the first element on the line, since everything after the first element is ignored.
   - If the first element on the line starts with //, the line is considered to be a comment and is ignored.
   - Each macro that you list in the text file must have a .mac extension.
3. Put this text file in the same location as the macros that it references.
4. In the Deployment Wizard, click the Configure menu on the Host Sessions window and select Server macro library... Check the 'Use a server macro library for this session' box and select Web server macro library.

5. Specify the fully qualified URL of the macro list that you created in Step 2, for example, http://servername/hod/macrolist.txt. Click OK.

When users open their sessions, they can use the Play Macro or Available Macros windows to see the macros specified in the list that you created for their session. These macros are available when users select Server library as their macro location. The Server library location is only available if you have configured the session to use a server macro library.

Deploying a server macro library to a shared drive

1. Put your macros in a shared directory on your network. Examples of valid directories include the following:
   - Absolute paths. Mapped network drive letters can also be used in the absolute path. Note that a server macro library should never point to a local drive.
   - Remote computer names or IP addresses are allowed as long as the user's computer is already remotely connected and authenticated to the computer that is sharing the directory. The following are two examples of paths to shared drive macro libraries:
     - `\\your_host\macro_library`, where *your_host* is the host name and *macro_library* is the macro directory.
     - `\\123.45.67.89\macro_library`, where *123.45.67.89* is the IP address of the host and *macro_library* is the macro directory.

   If you are configuring a macro library for more than one session, and each session uses its own set of macros, you will need to create a separate directory for each session.
2. In the Deployment Wizard Host Sessions window, select the session you wish to configure, click the Configure menu, and select Server macro library. Check the 'Use a server macro library for this session' box and select Shared drive macro library.
3. Specify the directory path that you set up in Step 1. Click OK.

When users open their sessions, they can use the Play Macro or the Available Macros windows to see a list of the macros in the directory. These macros are available when users select Server library as their macro location. The Server library location is only available if you have configured the session to use a server macro library.

Chapter 15. Modifying session properties dynamically

Host On-Demand sessions are defined by the administrator and retrieved by the Host On-Demand client when a user accesses a Host On-Demand HTML file. The session properties a user sees are fixed values and consist of a combination of the administrator's initial configuration and any user updates. However, there may be times when it would be useful with some HTML files, or with certain session properties, to dynamically set a value at the time that the HTML is accessed. This type of control allows you to set particular session property values based on information such as the IP address of the client or the time of day.

In order to dynamically set session properties at the time the HTML is accessed, the administrator must write a program that runs on the Web server and effectively modifies the HTML just before it is sent to the client. Even though the initial session properties are not defined in the HTML, Host On-Demand provides the capability to override many of the session properties in the HTML. These override values are always used by the client and take precedence over both the initial session properties setup by the administrator, as well as any updates for the property made by the user. The HTML override value is never stored, so the client will return to using prior settings for the property whenever the administrator removes the override. Also, the overridden property is locked so a user cannot change it.

There are many ways in which an administrator could write a program to dynamically set one or more session properties using the HTML overrides, such as using Java Server Pages (JSP), servlets, Perl, REXX, or Active Server Pages (ASP). This chapter takes you through a couple of examples that focus on common administrator issues. These examples are meant to demonstrate the syntax and technique of overriding particular properties. These mechanisms apply to whichever programming approach the administrator may choose.

Setting up the initial HTML

The initial HTML should be created using the Deployment Wizard, which will allow you to set up the features that are important to you, such as the size of the downloaded code and the functions available to your users. It will also help you by generating HTML that is correctly formatted for the Host On-Demand Java level you wish to support. The following sections describe the HTML parameters you will need to include. However, keep in mind that the exact format required for these parameters will vary depending on the format of the HTML, which, in turn, depends on the Host On-Demand Java level supported. Examples using both formats (Java 1 and Java 2/Auto Detect) are shown at the end of this chapter. Note that in Host On-Demand 7 and later, some of the HTML is generated using JavaScript, and HTML parameters are specified within a JavaScript array or using JavaScript document.write statements. Also, the format of the HTML varies according to the Java type (Java 1, Java 2, or Auto Detect) selected and whether the cached or download client is selected.

Setting the Code base

To set the code base when creating an HTML using the Deployment Wizard, do the following:

1. On the Additional Options window, click Advanced Options and go to the Other branch in the tree view.
2. Type the relative path /hod/ in the Code base field.
3. Save the HTML file to the default Host On-Demand publish directory *your_install_directory*\HOD.

The HTML file is now located in the same directory with the Host On-Demand's archive files.

Code base refers to the installed Host On-Demand publish directory and not the directory where Deployment Wizard files are published. Although you can enter a fully qualified URL in the Code base field, we strongly recommend that you enter the relative path /hod/ for the default publish directory when modifying session properties dynamically. If you enter a fully qualified URL, any users who specify the host name in a different manner than you specified as the Code base will not be able to access the files, even if the DNS entries resolve to the same IP address.

For more information about Code base and which files are created by the Deployment Wizard, refer to the Deployment Wizard chapter in the Host Access Client Package redbook on the IBM redbooks Web site at http://www.redbooks.ibm.com.

Overriding HTML parameters

There are several steps you must follow in order to dynamically set session properties (the examples shown later in this chapter will help clarify how some of these parameters should be specified):

1. **Enable HTML overrides**. By default, the client will ignore HTML overrides. To enable overrides, you will need to include an HTML parameter called EnableHTMLOverrides and set it to a value of true.
2. **List the sessions to be overridden**. Because there may be multiple sessions associated with an HTML, you will need to list which ones will be overridden. You will need to include an HTML parameter called TargetedSessionList, having a value of the exact names of the sessions that should accept overrides. The value should be a comma-separated list of session names, such as "Session1Name, Session2Name".
3. **Specify the override itself**. For each session property to be overridden, you will need to include an HTML parameter called the property name, with the value being the desired override. The value you specify will then apply to all sessions listed in your TargetedSessionList parameter. If you wish to only override a subset of the sessions in your TargetedSessionList, you can specify a value in the format of "Session1Name=value1, Session2Name=value2", for example.

Specific session properties that can be overridden

The following table describes the session properties that can be overridden and gives the acceptable values for each parameter:

Table 18. Session properties that can be overridden

Parameter name	Description	Valid values
Host	Host name or IP address of the target server. Appears as ″Destination address″ on property panels. Applies to all session types.	Host name or IP address.
HostBackup1	Host name or IP address of the backup1 server. Appears as ″Destination address″ of backup1on property panels. Applies to all session types.	Host name or IP address.
HostBackup2	Host name or IP address of the backup2 server. Appears as ″Destination address″ of backup2on property panels. Applies to all session types.	Host name or IP address.
Port	The port number on which the target server is listening. Appears as ″Destination port″ on property panels. Applies to all session types.	Any valid TCP/IP port number.
PortBackup1	The port number on which the backup1 server is listening. Appears as ″Destination port″ of backup1 on property panels. Applies to all session types.	Any valid TCP/IP port number.
PortBackup2	The port number on which the backup2 server is listening. Appears as ″Destination port″ of backup2 on property panels. Applies to all session types.	Any valid TCP/IP port number.
CodePage	The codepage of the server to which the session will connect. Appears as ″Host Code-Page″ on property panels. Applies to all session types except FTP.	The numeric portion (for example, 037) of the supported host codepage listed in the session property panel.

Table 18. Session properties that can be overridden (continued)

Parameter name	Description	Valid values
SessionID	The short name you want to assign to this session (appears in the OIA). It must be unique to this configuration. Appears as ″Session ID″ on property panels. Applies to all session types.	One character: A-Z.
LUName	The name of the LU or LU Pool, defined at the target server, to which you want this session to connect. Appears as ″LU or Pool Name″ on property panels. Applies to 3270 Display and 3270 Printer session types.	The name of an LU or LU Pool.
LUNameBackup1	The name of the LU or LU Pool, defined at the backup1 server, to which you want this session to connect. Appears as ″LU or Pool Name″ of backup1 on property panels. Applies to 3270 Display and 3270 Printer session types.	The name of an LU or LU Pool.
LUNameBackup2	The name of the LU or LU Pool, defined at the backup2 server, to which you want this session to connect. Appears as ″LU or Pool Name″ of backup2 on property panels. Applies to 3270 Display and 3270 Printer session types.	The name of an LU or LU Pool.
WorkstationID	The name of this workstation. Appears as ″Workstation ID″ on property panels. Applies to 5250 Display and 5250 Print session types.	A unique name for this workstation.

Table 18. Session properties that can be overridden (continued)

Parameter name	Description	Valid values
ScreenSize	Defines the number of rows and columns on the screen. Appears as ″Screen Size″ on property panels. Applies to 3270 Display, 5250 Display, and VT Display session types.	• value=rows x columns • 2=24x80 (3270, 5250, VT) • 3=32x80 (3270) • 4=43x80 (3270) • 5=27x132 (3270, 5250) • 6=24x132 (VT) • 7=36x80 (VT) • 8=36x132 (VT) • 9=48x80 (VT) • 10=48x132 (VT) • 11=72x80 (VT) • 12=72x132 (VT) • 13=144x80 (VT) • 14=144x132 (VT) • 15=25x80 (VT) • 16=25x132 (VT)
SLPScope	Service Location Protocol (SLP) Scope. Appears as ″Scope″ under ″SLP Options″ on property panels. Applies to 3270 Display, 3270 Printer, 5250 Display, and 5250 Printer session types.	Contact your administrator to get the correct value for this field.
SLPAS400Name	Connects a session to a specific iSeries. Appears as ″AS/400 Name (SLP)″ on property panels. Applies to 5250 Display and 5250 Printer session types.	The fully-qualified SNA CP name (for example, USIBMNM.RAS400B).
SSLCertificateSource	The certificate can be kept in the client's browser or dedicated security device, such as a smart card; or, it can be kept in a local or network-accessed file. Appears as ″Certificate Source″ on property panels. Applies to 3270 Display, 3270 Printer, 5250 Display, 5250 Printer, and VT Display session types.	The value is SSL_CERTIFICATE_IN_CSP for a certificate in a browser or security device. The value is SSL_CERTIFICATE_IN_URL for a certificate in a URL or file.

Table 18. Session properties that can be overridden (continued)

Parameter name	Description	Valid values
SSLCertificateURL	Specifies the default location of the client certificate. Appears as ″URL or Path and Filename″ in property panels. Applies to 3270 Display, 3270 Printer, 5250 Display, 5250 Printer, and VT Display session types.	The URL protocols you can use depend on the capabilities of your browser. Most browsers support HTTP, HTTPS, FTP, and FTPS.
FTPUser	Specifies the user ID the session uses when connecting to the FTP server. Appears as ″User ID″ on property panels. Applies to FTP session types.	A valid user ID.
FTPPassword	Specifies the password the session uses when connecting to the FTP server. Appears as ″Password″ on property panels. Applies to FTP session types.	A valid password.
UseFTPAnonymousLogon	Enables the session to log in to an FTP server using anonymous as the user ID. Appears as ″Anonymous Login″ on property panels. Applies to FTP session types.	Yes or No.
FTPEmailAddress	Specifies the e-mail address to use when connecting to the FTP server while using Anonymous Login. Appears as ″E-mail Address″ on property panels. Applies to FTP session types.	A valid e-mail address.
CICSInitialTransEnabled	Enables an initial transaction to be started when a CICS Gateway session is established.	true or false
CICSInitialTrans	Specifies the name of the initial transaction to be started upon connection to a CICS host. Applies to CICS Gateway sessions only. The CICSInitialTransEnabled parameter must be set to true for the specified transaction to be started.	Valid transaction identifiers are strings of between 1 and 128 characters. The string identifies the initial transaction and any parameters to be run upon connection to the server. The first four characters, or the characters up to the first blank in the string are taken as the transaction. The remaining data is passed to the transaction on its invocation.

Table 18. Session properties that can be overridden (continued)

Parameter name	Description	Valid values
Netname	The name of the terminal resource to be installed or reserved. If this field is blank, the selected terminal type is not predictable. Applies to CICS sessions only.	A valid terminal resource name.

Any errors encountered in processing the HTML parameters is displayed in the Java console.

Example #1: Overriding the LU name based on the client's IP address

Administrators may want to avoid specifying LU names directly in session definitions. This example shows a simple way of using the IP address of the client to look up an LU name listed in a text file and use it as an override value in a session.

This example is written using JSP. The Deployment Wizard was used to create an HTML file that contains two sessions named 3270 Display and 5250 Display. Note that in Host On-Demand 7 and later, some of the HTML is generated using JavaScript, and HTML parameters are specified within a JavaScript array or using JavaScript document.write statements. Also, the format of the HTML varies according to the Java type (Java 1, Java 2, or Auto Detect) selected and whether the cached or download client is selected. In this example, a Java 1 cached client was selected.

A file (c:\luname.table) is read that contains IP address/LU name pairs. The IP address of the client is used to look up the proper LU name, which is overridden in the ″3270 Display″ session. See the comments in the example for more detail. The lines added to the Deployment Wizard output are displayed in **bold**.

```
<!doctype html public "-//W3C//DTD HTML 3.2 Final//EN">
<%
//  Read the luname.table file into a properties variable.
//  The luname.table file contains lines in the following format:
//    ipaddress=luname
Properties lunames = new Properties();
lunames.load(new FileInputStream("c:\\luname.table"));
%>
<!-- HOD WIZARD HTML -->
<HTML>
<HEAD>
<META http-equiv="content-type" content="text/html; charset=UTF-8">
<!-- TITLE Begin -->
<TITLE>Example 1 page title</TITLE>
<!-- TITLE End -->
<!-- SUMMARY Begin -->
<!--
Configuration Model
 What configuration model would you like to use?
 -HTML-based model
Host Sessions
 -3270 Display
 -5250 Display
Additional Options
 -Cached = Cached client
 -Java Type = java1
Disable Functions
Preload Options
 -5250 Sessions = True
 -Change Session Properties = True
 -3270 Sessions = True
```

```
Cached Client/Web Start Options
 Basic Options
 -Debug = False
 -Height (in pixels) = 250
 -Width  (in pixels) = 550
 Upgrade Options
 -Percent of users who can upgrade by default = 100
 -Prompt user (user decides foreground or background)
Advanced Options
 HTML parameters
 -None
 Code base
 - /hod/
 HTML templates
 -Default
 Problem determination
 -Debug = False
 User updates
 -Persist user updates? = True
 Appearance
 -Standard Host On-Demand Client
 Applet size
 -Autosize to browser
 Session Manager API
 -Enable Session Manager JavaScript API = False
 Server connection
 Language
 -Locale = Use the system Locale
 Maximum sessions
 - 26
-->
<!-- SUMMARY End -->
</HEAD>

<BODY BACKGROUND="/hod/hodbkgnd.gif">
<CENTER>
<IMG src="/hod/hodlogo.gif" ALT="hodlogo.gif">
<P>


<SCRIPT LANGUAGE="JavaScript">
function writeAppletParameters()
{
   document.write("");
}
</SCRIPT>


<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CachedJ1.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
var hod_Height='80%';
var hod_Width='80%';

codebase='/hod/';
installer='/hod/Installer.html';

document.write('<APPLET CODEBASE="/hod/" ARCHIVE="CachedAppletSupporter.jar"
   MAYSCRIPT NAME="HODApplet" CODE="com.ibm.eNetwork.HOD.cached.appletloader.CachedAppletLoader"
   WIDTH="'+hod_Width+'" HEIGHT="'+hod_Height+'">');
document.write('<PARAM NAME="Cabinets"       VALUE="CachedAppletSupporter.cab">');
document.write('<PARAM NAME="CachedClient"   VALUE="true">');
document.write('<PARAM NAME="ParameterFile"  VALUE="HODData\\Example1\\params.txt">');
document.write('<PARAM NAME="JavaScriptAPI"  VALUE="false">');

// The next 2 lines are required in order to override session properties.
//  The first line turns on the processing for this function and does not
//  need to be modified.  The second line identifies the sessions that you
//  want to change.  In this example, there are 2 sessions identified
//  named: "3270 Display" and "5250 Display".

document.write('<PARAM NAME="EnableHTMLOverrides" VALUE="true">');
document.write('<PARAM NAME="TargetedSessionList"
    VALUE="3270 Display,5250 Display">');

// The following line changes the LUName session parameter for the session named
//  "3270 Display".  In this example, the LUName is being set to the value
//  contained in the c:\luname.table for the IP address of the client.
//  When you are initially testing your changes, you may want to use a constant
//  value to verify that the syntax is correct before you insert your
//  calculations.
```

```
document.write('<PARAM NAME="Luname" VALUE="3270
Display=<%=lunames.get(request.getRemoteAddr())%>">');

writeAppletParameters();
document.write("</APPLET>");
</SCRIPT>

<P>
<SCRIPT LANGUAGE="JavaScript">
var hod_AppName='';
var hod_Preloadlist='HABASE;HODBASE;HODIMG;HACP;HAFNTIB;HAFNTAP;HA3270;HODCFG;HA5250';
var hod_Debugcomponents='false';
var hod_Debugcachedclient='false';
var hod_Upgradepromptresponse='Prompt';
var hod_Upgradepercent='100';
var hod_Framewidth='550';
var hod_Frameheight='250';

function isBookmark(mySearch) {
  if (mySearch.length < 2) {
    return false;
  } else {
    return (mySearch.toLowerCase().indexOf('launch=') != -1);
  }
}

if (hod_AppName == '') {
  if (isBookmark(window.location.search.substring(1)))
    hod_AppName = 'com.ibm.eNetwork.HOD.SessionLauncher';
  else
    hod_AppName = 'com.ibm.eNetwork.HOD.HostOnDemand';
}

function getHODFrame() {
  return self;
}

document.write('<APPLET CODEBASE="/hod/" ARCHIVE="CachedAppletSupporter.jar"
  MAYSCRIPT NAME="CachedAppletSupporter"
  CODE="com.ibm.eNetwork.HOD.cached.appletsupport.CachedAppletSupportApplet"
  WIDTH="2" HEIGHT="2">');
document.write('<PARAM NAME="Cabinets"
                                      VALUE="CachedAppletSupporter.cab">');
document.write('<PARAM NAME="DebugComponents"
                                      VALUE="'+hod_Debugcomponents+'">');
document.write('<PARAM NAME="PreloadComponentList"
                                      VALUE="'+hod_Preloadlist+'">');
document.write('<PARAM NAME="DebugCachedClient"
                                      VALUE="'+hod_Debugcachedclient+'">');
document.write('<PARAM NAME="CachedClientSupportedApplet"
                                      VALUE="'+hod_AppName+'">');
document.write('<PARAM NAME="InstallerFrameWidth"
                                      VALUE="'+hod_Framewidth+'">');
document.write('<PARAM NAME="InstallerFrameHeight"
                                      VALUE="'+hod_Frameheight+'">');
document.write('<PARAM NAME="UpgradePromptResponse"
                                      VALUE="'+hod_Upgradepromptresponse+'">');
document.write('<PARAM NAME="UpgradePercent"
                                      VALUE="'+hod_Upgradepercent+'">');
document.write("</APPLET>");
</SCRIPT>


</CENTER>
</BODY>
</HTML>
```

This example uses a cached Java 2 page to start from with the needed changes for HTML overrides in bold. When the Deployment Wizard is used to generate a cached Java2 page it generates the following files:

- Example1.html
- z_Example1.html
- Example_J2.html

A Macintosh client makes use of the Example_J2.html page.

A file (c:\luname.table) is read that contains IP address/LU name pairs. The IP address of the client is used to look up the proper LU name, which is overridden in the ″3270 Display″ session. See the comments in the example for more detail. The lines added to the Deployment Wizard output are displayed in **bold**.

```
<!doctype html public "-//W3C//DTD HTML 3.2 Final//EN">
<%
//  Read the luname.table file into a properties variable.
//  The luname.table file contains lines in the following format:
//    ipaddress=luname
Properties lunames = new Properties();
lunames.load(new FileInputStream("c:\\luname.table"));
%>
<HTML>
<HEAD>
<META http-equiv="content-type" content="text/html; charset=UTF-8">
<!-- TITLE Begin -->
<TITLE>Example1 page title</TITLE>
<!-- TITLE End -->
<!-- SUMMARY Begin -->
<!--
Configuration Model
 What configuration model would you like to use?
 -HTML-based model
Host Sessions
 -3270 Display
 -5250 Display
Additional Options
 -Cached = Cached client
 -Java Type = java2
Disable Functions
Preload Options
 -5250 Sessions = True
 -Change Session Properties = True
 -3270 Sessions = True
Cached Client/Web Start Options
 Basic Options
 -Debug = False
 -Height (in pixels) = 250
 -Width  (in pixels) = 550
 Upgrade Options
 -Percent of users who can upgrade by default = 100
 -Prompt user (user decides foreground or background)
Advanced Options
 HTML parameters
 -None
 Code base
 - /hod/
 HTML templates
 -Default
 Problem determination
 -Debug = False
 User updates
 -Persist user updates? = True
 Appearance
 -Standard Host On-Demand Client
 Applet size
 -Autosize to browser
 Session Manager API
 -Enable Session Manager JavaScript API = False
 Server connection
 Language
 -Locale = Use the system Locale
 Maximum sessions
 - 26
-->
<!-- SUMMARY End -->
</HEAD>

<BODY BACKGROUND="/hod/hodbkgnd.gif">
<CENTER>
<IMG src="/hod/hodlogo.gif" ALT="hodlogo.gif">
<P>


<SCRIPT LANGUAGE="JavaScript">
function writeAppletParameters()
{
   return "";
```

```
}
</SCRIPT>


<SCRIPT LANGUAGE="JavaScript" SRC="/hod/HODVersion.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CommonJars.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CommonParms.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CommonJ2Parms.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
var db = parent.location;
var hod_Locale = '';
var hod_AppName ='';
var hod_AppHgt = '340';
var hod_AppWid = '550';
var hod_CodeBase = '/hod/';
var hod_Comps = 'HABASE;HODBASE;HODIMG;HACP;HAFNTIB;HAFNTAP;HA3270;HODCFG;HA5250';
var hod_Archs = 'habasen.jar,hodbasen.jar,hodimg.jar,hacp.jar,hafntib.jar,hafntap.jar,
                 ha3270n.jar,hodcfgn.jar,ha5250n.jar';

var hod_URL = new String(window.location);
var hod_DebugOn = false;

// put cached client installation applet parameters here
var hHod_AppletParams = new Array;
hHod_AppletParams[0] = '<PARAM NAME="DebugCachedClient" VALUE="false">';
hHod_AppletParams[1] = '<PARAM NAME="ShowDocument"      VALUE="_parent">';
hHod_AppletParams[2] = '<PARAM NAME="CachedClient"      VALUE="true">';
hHod_AppletParams[3] = '<PARAM NAME="ParameterFile"     VALUE="HODData\\Example1\\params.txt">';
hHod_AppletParams[4] = '<PARAM NAME="JavaScriptAPI"     VALUE="false">';
hHod_AppletParams[5] = '<PARAM NAME="BookmarkPage"      VALUE="Example1.html">';

// The next 2 lines are required in order to override session properties.
//  The first line turns on the processing for this function and does not
//  need to be modified.  The second line identifies the sessions that you
//  want to change.  In this example, there are 2 sessions identified
//  named: "3270 Display" and "5250 Display".

hHod_AppletParams[6]='<PARAM NAME="EnableHTMLOverrides" VALUE="true">';
hHod_AppletParams[7]='<PARAM NAME="TargetedSessionList" VALUE="3270 Display,5250 Display">';

// The following line changes the LUName session parameter for the session named
//  "3270 Display".  In this example, the LUName is being set to the value
//  contained in the c:\luname.table for the IP address of the client.
//  When you are initially testing your changes, you may want to use a constant
//  value to verify that the syntax is correct before you insert your
//  calculations.
hHod_AppletParams[8]='<PARAM NAME="Luname" VALUE="3270
                      Display=<%=lunames.get(request.getRemoteAddr())%>">';

//hHod_AppletParams[x] = '<PARAM NAME="DebugCode"             VALUE="65535">';


var pg = buildJ2Page(db);
pg += writeAppletParameters();
pg += '</APPLET>';
if(hod_DebugOn) alert('J2 page complete, result = \n' + pg);
document.write(pg);
</SCRIPT>

</CENTER>
</BODY>
</HTML>
```

Example #2: Allowing the user to specify the host to connect to using an HTML form

Administrators may also want to use HTML forms to specify override values rather than calculating them. The following example displays a simple form for entry of a host name. The form posts to a JSP program which uses the host name specified in the form to override the host name in the 3270 Session.

This example is written using JSP. The Deployment Wizard was used to create an HTML file that contains two sessions named ″3270 Display″ and ″5250 Display.″ Note that in Host On-Demand 7 and later, some of the HTML is generated using

JavaScript, and HTML parameters are specified within a JavaScript array or using JavaScript document.write statements. Also, the format of the HTML varies according to the Java type (Java 1, Java 2, or Auto Detect) selected and whether the cached or download client is selected. In this example, a Java Detect download client was selected.

When using forms, the form data needs to be retained across requests to the program. This is because Host On-Demand HTML files reload themselves for Java detection and for bookmarking support when using configuration server-based model pages. If Java 1 is selected and bookmarking support is disabled if using the configuration server-based model, the page will not need to reload and there is no need to retain the form data. This example uses a JSP session to store the form data across reloads.

Here is a simple HTML form that allows for entry of a host name. The form posts to the JSP program (example2.jsp):

```
<form method="POST"  action="hod/example2.jsp">
Hostname <input name="form.hostname"><br>
<input type="submit">
</form>
```

Here is the modified output from the Deployment Wizard. See the comments in the example for more detail. The lines added to the Deployment Wizard output are displayed in **bold**.

```
<HTML>
<%
// Get a session or create if necessary and store the hostname
// entered in the form in the session.
HttpSession session = request.getSession(true);
String hostname = request.getParameter("form.hostname");
if (hostname!=null) {
session.putValue("session.hostname", hostname);
}
%>
<!-- HOD WIZARD HTML -->
<!-- Deployment Wizard Build : 8.0.0-B20030605 -->
<HEAD>
<META http-equiv="content-type" content="text/html; charset=UTF-8">
<TITLE>Example 2 page title</TITLE>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CommonJars.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/HODJavaDetect.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CommonParms.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript">

//---- Start JavaScript variable declarations ----//
var hod_Locale = '';
var hod_jsapi=false;
var hod_AppName ='';
var hod_AppHgt = '80%';
var hod_AppWid = '80%';
var hod_CodeBase = '/hod/';
var hod_FinalFile = 'z_example2.html';
var hod_JavaType = 'detect';
var hod_Obplet = '';
var hod_jars = 'habasen.jar,hodbasen.jar,hodimg.jar,hacp.jar,hodsignn.jar,ha3270n.jar,
                hodcfgn.jar,ha5250n.jar';

var hod_URL = new String(window.location);
var hod_DebugOn = false;
var hod_SearchArg = window.location.search.substring(1);

var hod_AppletParams = new Array;
hod_AppletParams[0] = '<PARAM NAME="ParameterFile" VALUE="HODData\\example2\\params.txt">';
hod_AppletParams[1] = '<PARAM NAME="ShowDocument"  VALUE="_parent">';
hod_AppletParams[2] = '<PARAM NAME="JavaScriptAPI" VALUE="' + hod_jsapi + '">';
hod_AppletParams[3] = '<PARAM NAME="PreloadComponentList" VALUE="HABASE;HODBASE;HODIMG;
                                                                HACP;HAFNTIB;HAFNTAP;
                                                                HA3270;HODCFG;HA5250">';

// The next 2 lines are required in order to override session properties.
```

```
// The first line turns on the processing for this function and does not
// need to be modified. The second line identifies the sessions that you
// want to change. In this example, there are 2 sessions identified
// named: "3270 Display" and "5250 Display".
// Be careful to increment the array index correctly.

hod_AppletParams[4] = <PARAM NAME="EnableHTMLOverrides" VALUE="true">;
hod_AppletParams[5] = <PARAM NAME="TargetedSessionList" VALUE="3270 Display,5250 Display">;

// The following line changes the Host or Destination Address session parameter
// for the session named "3270 Display". In this example, the Host is being set
// to the value saved in the JSP session from the HTLM form.
// When you are initially testing your changes, you may want to use a constant
// value to verify that the syntax is correct before you insert your
// calculations.
// Here we override the host for the 3270 session to the value saved in the
// jsp session from the html form.

hod_AppletParams[6] = <PARAM NAME="Host" VALUE="3270
                       Display=<%=session.getValue("session.hostname")%>">;

//hod_AppletParams[x] = '<PARAM NAME="DebugCode"     VALUE="65535">';

//---- End JavaScript variable declarations ----//

function getHODMsg(msgNum) {
  return HODFrame.hodMsgs[msgNum];
}

function getHODFrame() {
  return HODFrame;
}

var lang = detectLanguage(hod_Locale);
document.writeln('<FRAMESET cols="*,10" border=0 FRAMEBORDER="0">');
document.writeln('<FRAME    src="/hod/hoddetect_' + lang + '.html" name="HODFrame">');
document.writeln('</FRAMESET>');

</SCRIPT>
</HEAD>
</HTML>
```

Chapter 16. Configuring Host On-Demand on zSeries

This chapter concentrates on two specific scenarios for configuring Host On-Demand on a zSeries system:

- Installing, configuring, and using the Host On-Demand configuration servlet in the WebSphere Application Server environment to communicate between Host On-Demand clients and the Host On-Demand Service Manager.
- Setting up separate read/write private and publish directories.

These configuration scenarios have several purposes:

- They provide instructions for common zSeries configuration tasks.
- They gather information on multiple products in one place, making it easier for users to perform complex configuration tasks.
- They show how Host On-Demand interacts with other WebSphere products, such as WebSphere Application Server.

See the product installation documentation (found in the Program Directory) for detailed instructions on setting up Host On-Demand on zSeries. For more information on the products involved in these configuration scenarios, see the product documentation, IBM Redbooks, and other product-related material.

Installing and configuring the Host On-Demand configuration servlet

By default, the Host On-Demand clients use port 8999 to access configuration information from the Service Manager. If any of your clients are outside the firewall, the firewall administrator needs to open port 8999 both internally and externally. However, with Host On-Demand you can avoid opening this port by customizing your clients to use the configuration servlet to access configuration information. It can be configured to run either from the WebSphere Application Server HTTP server plug-in or from the WebSphere Application Server Web container.

The steps required to configure Host On-Demand are as follows:

1. Set up the zSeries system and install Host On-Demand, WebSphere Application Server 4.0.1, and the IBM HTTP server.
2. Modify the HTTP server configuration file.
3. Set up the HTTP server environment variables.
4. Decide whether you want the configuration servlet to run from the WebSphere Application Server version 4.0 plug-in or the WebSphere Application Server version 4.0 Web container, and install it accordingly.
5. Enable clients to use the configuration servlet.
6. Restart the HTTP server and the Host On-Demand Service Manager.
7. Verify that the configuration servlet is enabled.

If you receive the following error when starting the Service Manager:

```
remote.Server. : Server Socket Contructor Failed:EDC81151 Address already in use.
***Error - Failed to start Service Manager on port 8999
```

check in the BPXPRMxx member of SYS1.PARMLIB or the PARMLIB (which contains the BPXPRMxx member) to see if the INADDRANYPORT and INADDRANYCOUNT parameters have a port range that includes 8999. If so, change the port range to exclude 8999 for Host On-Demand. Refer to *MVS Initialization and Tuning Reference, SC28–1752* for more information about BPXPRMxx and the INADDRANYPORT and INADDRANYCOUNT parameters.

Set up the zSeries system

Before you start configuring Host On-Demand, you need to set up the zSeries system.

1. Verify that the following are installed:
   - OS/390 V2R10, z/OS V1.1 or later
   - The Communication Server package that is shipped with the operating system
2. Install WebSphere Application Server version 4.0.1 and run the installation verification program (IVP). See "WebSphere Application Server 4.0.1 requirements" for more information.
3. Install the IBM HTTP server version 5.3.
4. Install Host On-Demand 8.

WebSphere Application Server 4.0.1 requirements

WebSphere Application Server version 4.0.1 has the following requirements:

- Workload management (WLM). See the IBM Redbook *Prepare OS/390 for WebSphere Enterprise Edition* (part number SG24–5685–00), available at www.redbooks.ibm.com. Follow the instructions in chapter 2 to set up a monoplex and chapter 3 to set up workload management and switch into GOAL mode.
- System logger. Follow the setup instructions in chapter 4 of *Prepare OS/390 for WebSphere Enterprise Edition*.
- Resource recovery service (RRS). Follow the setup instructions in chapter 5 of *Prepare OS/390 for WebSphere Enterprise Edition*.
- IBM DB2 relational database, 7 release 1. See the *DB2 UDB for OS/390 and zOS V7 Installation Guide* (part number GC26–9936–01). Follow the instructions in the "Installing, migrating, and updating system parameters" and "Installing the DB2 subsystem" chapters to set up DB2.
- LDAP. (WebSphere Application Server customization leads you through the required LDAP configuration steps.)
- Java 1.3

After installing WebSphere Application Server, run the IVP. In this scenario, you will be using one of the application servers (BBOASR2) that is set up as part of the IVP. For more information on installing WebSphere Application Server and running the IVP, see the *WebSphere Application Server version 4.0.1 for z/OS and OS/390 Installation and Customization Guide* (part number GA22–78834–02). Follow all instructions for running the installation CLIST and configuration jobs, and complete the IVP.

Tips for configuring WebSphere Application Server: The following tips can help you to successfully configure WebSphere Application Server 4.0.1:

- Increase the BP32K buffer pools in DB2 to at least 100.
- Set up 32K temporary work files in DB2.
- Run WLM in goal mode. To find out whether your zSeries system is running WLM in this mode, enter the following command from the z/OS or OS/390 system console:

  ```
  d wlm,systems
  ```

  If the system is not in goal mode, enter the following command:

  ```
  modify wlm,mode=goal
  ```

- Make sure that the following WLM application environments are available:
  - CBSYSMGT
  - CBNAMING
  - CBINTFRP
  - BBOASR2

  To view the available environments, issue the following command from the z/OS or OS/390 system console:

  ```
  display wlm,applenv=*
  ```

  If one of the previous environments is not available (for example, CBSYSMGT), issue the following command:

  ```
  vary wlm,applenv=CBSYSMGT,resume
  ```

- Before running the job BBOCBGRT, define the DSNJDBC plan by running the DB2 job DSNTJJCL.
- Before running the job BBOLD2DB, verify that the LDAP module is included in the link list and is APF authorized. (For example, the module was 'GLD.SGLDLNK' on our test system.)
- If you have problems bringing up the BBOASR2 application server defined in the WAS 4.0.1 IVP, do the following:
  - Add /usr/lib to the classpaths for the BBOASR2 server and the CBSYSMGT server.
  - Remove /usr/lib from the classpath for the CBNAMING server.
- Verify that the host.default_host.alias value in your was.conf file is correct for your system.

Modify the HTTP server configuration file

After you have verified that the system has been set up correctly, add the following lines to the HTTP Web server config file /etc/httpd.conf:

- The servlet initialization statement:

  ```
  ServerInit /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plugin.so:init_exit
     /usr/lpp/WebSphere,/config_dir/was.conf
  ```

  where *config_dir* is the directory where the was.conf file is stored. This statement must be on one line in the /etc/httpd.conf file. You also need to verify that the host.default_host.alias value in the was.conf file is correct for your system.

- The service statement for the configuration servlet:

  ```
  Service/HodConfig/*
       /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plugin.so:service_exit
  ```

This statement must be on one line in the /etc/httpd.conf file.

Set up the HTTP server environment variables

Set the values of the following environment variables in the file /etc/httpd.envars:

Table 19. zSeries HTTP server environment variables

Environment variable	Value
JAVA_HOME	Set this variable to the location of the SDK home directory. For example: /usr/lpp/java/IBM/1.3
NLSPATH	Add the following directory to this variable: /usr/lpp/WebSphere/WebServerPlugIn/msg/%L/%N
LIBPATH	Add the following directory to this variable: /usr/lpp/WebSphere/wc/lib
CLASSPATH	Add the following directory to this variable: /usr/lpp/WebSphere/wc/lib

See the WebSphere Application Server 4.0.1 IVP instructions for detailed explanations of these environment variables.

Install the configuration servlet

You have two options for installing the configuration servlet:

- Install the configuration servlet to be run from the WebSphere Application Server version 4.0.1 Web container. Use this option if you are setting up a new WebSphere Application Server environment or would like to migrate an existing environment to the new Web container. See "Installing and running the configuration servlet from the Web container" for instructions.
- Install the configuration servlet to be run from the WebSphere Application Server version 4.0 HTTP server plug-in (which is part of the version 4.0.1 plug-in). Use this option if you would like to preserve an environment previously set up for version 3.5 of WebSphere Application Server (for example, for migration purposes). See "Installing and running the configuration servlet from the plug-in" on page 142 for instructions.

Because you cannot configure both the version 4.0 plug-in and the version 4.0.1 Web container in the same Web server, you can only select one of these options. For more detailed information on how to select an installation option, see the white paper *WebSphere Application Server V4.0 and V4.0.1 for zOS and OS/390 Configuring Web Applications* (WP100238), available from http://www.ibm.com/support/techdocs.

Installing and running the configuration servlet from the Web container

Installing and running the configuration servlet from the Web container is a two-part process:

1. Use the WebSphere Application Assembly Tool (AAT) to configure the cfgservlet.ear file (which contains the configuration servlet) for the Web container.
2. Use the WebSphere Administration Tool to install the cfgservlet.ear file in the Web container.

Configuring the cfgservlet.ear file with the AAT: Use the AAT to install the configuration servlet as follows:

1. Download the file /usr/lpp/HOD/hostondemand/lib/cgfservlet.ear in binary to your Windows system.
2. If it is not already installed on your system, download the AAT from the Web site http://www.ibm.com/software/webservers/appserv/. Click on Download and scroll for the link to WebSphere Application Server V4.0 for z/OS and OS/390 downloads. Select the AAT and follow the instructions to download and install it.
3. Launch the AAT.
4. Import the cgfservlet.ear file.
5. In the AAT window, expand Host On-Demand Configuration Servlet > Web Apps > cfgservlet.war > Web Components.
6. Select HOD Config Servlet, click the right mouse button, and select Modify.
7. Select the Parameters tab.
8. Modify the following parameters:

   ShowStats
   True

   Trace True

   ConfigServerPort
   8999 (the default). If you want the configuration servlet to use a different port than the default, change this value.
9. Save your changes.
10. Select Host On-Demand Configuration Servlet, click the right mouse button, and select Validate.
11. Select Host On-Demand Configuration Servlet > Deploy to prepare the application for export.
12. Select Host On-Demand Configuration Servlet > Export to regenerate the cfgservlet.ear file.

Install the cfgservlet.ear file with the WebSphere Application Server Administration Tool: To install the cfgservlet.ear file in the Web container, do the following:

1. Download and install the most current version of the WebSphere Application Server Administration Tool for zSeries from your server's WebSphere Application Server /bin directory (for example, /usr/lpp/WebSphere/bin/bboninst.exe).
2. Launch the WebSphere Application Server Administration Tool for zSeries.
3. Create a new conversation named HODConfig Servlet and save it.
4. Expand HODConfig Servlet > Sysplexes > *sysplex* > J2EE Servers > BBOARS2, where *sysplex* is the name of your sysplex.
5. Click the right mouse button on BBOARS2 and select Install J2EE Application.
6. Select the cfgservlet.ear file (use the Browse button if necessary).
7. Click Set the Default JNDI Path, then click OK.
8. Expand BBOASR2 > J2EEApplications. You should see Host On-Demand Configuration Servlet.
9. Select HODConfig Servlet, click the right mouse button, and select Validate.
10. Select HODConfig Servlet, click the right mouse button, and select Commit to commit the conversation.
11. Select HODConfig Servlet, click the right mouse button, and select Complete, All Tasks. Click Yes.

12. Use the z/OS or OS/390 system console to verify that the BBOARS2 server is running.
13. Select HODConfig Servlet, click the right mouse button, and select Activate. Click Yes. Be aware that this step can take some time. Wait for a message that the conversation has been activated.

Installing and running the configuration servlet from the plug-in

To install and run the configuration servlet from the plug-in, add the following to the */config_dir*/was.conf file (where *config_dir* is the directory where the configuration file is located). Optionally, you can change the default values of the ConfigServerPort, ShowStats and Trace parameters.

```
# ================================================================ #
#
#   The following defines the HOD Configuration Servlet
#
# ================================================================ #
deployedwebapp.HOD.host=default_host
deployedwebapp.HOD.rooturi=/HODConfig
deployedwebapp.HOD.classpath=/usr/lpp/HOD/hostondemand/HOD
     :/usr/lpp/HOD/hostondemand/lib/cfgsrvlt.jar
     :/usr/lpp/HOD/hostondemand/HOD/com/ibm/eNetwork/HODUtil/services/remote
     :/usr/lpp/HOD/hostondemand/HOD/com/ibm/eNetwork/HOD
deployedwebapp.HOD.documentroot=/usr/lpp/HOD/hostondemand/lib
deployedwebapp.HOD.autoreloadinterval=100000
webapp.HOD.jspmapping=*.jsp
webapp.HOD.jspmapping=*.jhtml
webapp.HOD.filemapping=/
webapp.HOD.jsplevel=1.1
webapp.HOD.servlet.HODConfigServlet.code
      =com.ibm.eNetwork.HODUtil.services.remote.HODCfgServlet
webapp.HOD.servlet.HODConfigServlet.servletmapping=/HODConfig
webapp.HOD.servlet.HODConfigServlet.initargs=ConfigServerPort=8999,
       ShowStats=true,Trace=false
webapp.HOD.servlet.HODConfigServlet.autostart=true
#####################################################################
```

The three lines following the `deployedwebapp.HOD.classpath` parameter, the line following the `webapp.HOD.servlet.HODConfigServlet.code` parameter, and the line following the `webapp.HOD.servlet.HODConfigServlet.initargs` parameter must be on the same line as the parameter in the actual was.conf file. To improve performance, set the Trace parameter to false.

If you changed the default configuration server port

If you changed the default value of the ConfigServerPort parameter while installing the configuration servlet in the plug-in or the Web container, you need to update the port number in the NSMprop and config.properties.ascii files.

- Add the following line to the /usr/lpp/HOD/hostondemand/private/NSMprop file:

  `CONFIGSERVER_PARMS = %INSTALL_PATH% portnumber`

  where *portnumber* is the new configuration servlet port.
- On a machine that supports ASCII, create a file called config.properties (if it is not already present) and add the following line:

  `ConfigServerPort=portnumber`

  Upload the config.properties file in binary format to the Host On-Demand publish directory on the zSeries system and save it as /usr/lpp/HOD/hostondemand/HOD/config.properties.ascii.

Enable clients to use configuration servlet

You can enable all clients to use the configuration servlet, or you can limit access to specific clients.

- To enable access for all clients, do the following:
  1. On a machine that supports ASCII, create a file called config.properties (if it is not already present) and add the following line:

     ```
     ConfigServerURL=http://server_name/HODConfig/HODConfig/hod
     ```

     where *server_name* is the name of the Host On-Demand server.
  2. Upload the config.properties file in binary format to the Host On-Demand publish directory on the zSeries system and save it as /usr/lpp/HOD/hostondemand/HOD/config.properties.ascii.
- To enable access only for specific clients, do the following:
  1. If it is not already installed, download and install the Deployment Wizard. See "Installing the Deployment Wizard" on page 70 for instructions.
  2. In the HTML model, use the Deployment Wizard to create HTML files and enable the configuration servlet. Set the following parameters in the Additional Parameters window:

     Name ConfigServerURL

     Value /HODConfig/HODConfig/hod

     In the configuration server or combined models, click Server Connection Options on the Additional Options window.
  3. Save the files generated from the Deployment Wizard as a Zip file.
  4. Use FTP to transfer the Zip file to the zSeries system.
  5. Use the DWunzip tool to install the HTML files on the zSeries system. See the online help topic Using DWunzip for more information on how to use this tool.

Verify that the configuration servlet is enabled

Finally, you need to verify that the configuration servlet has been enabled. Do the following:

1. Restart the HTTP server and the Host On-Demand Service Manager.
2. Bring up the HODMain.html file (if you enabled all clients to use configuration servlet) or your own custom Host On-Demand HTML file.
3. Verify that the configuration servlet is enabled by doing one or both of the following:
   - If you set the ShowStats parameter to True when you were installing the cfgservlet.ear file (as described in "Install the configuration servlet" on page 140), you can invoke the ShowStats function to test whether the servlet is running. Specify the following URL in your Web browser:

     ```
     http://server_name/servlet_location/HodConfig/info
     ```

     where *server_name* is the name of the zSeries server and *servlet_location* is the directory in which the configuration servlet is installed (in this example, HODConfig).

     If the configuration servlet is running, the browser window displays statistics gathered from the configuration servlet. The page shows the configuration servlet start time, address, ConfigServerPort, and other information about the

servlet. To verify whether the servlet is running, check to see if any POST requests have been processed, if any buffers have been created, and if data has been sent to and received by the Service Manager. Sample statistics from an active configuration servlet are shown in the following excerpt from the statistics HTML file:

```
Servlet Statistics

   Server Information = WebSphere Application Server for OS/390/4.1
   Servlet ID = 0
   152 POST request have been processed. The largest request
      contained 10640 bytes of data.
   The buffer pool currently contains 1 entries.
   A total of 9 buffers have been created.
   A total of 98344 bytes have been transfered to the Service Manager.
   A total of 100140 bytes have been received from the Service Manager.
```

- If you set the Trace parameter to True when you were installing the cfgservlet.ear file (as described in "Install the configuration servlet" on page 140), you can invoke the Trace function to test whether the servlet is running. Specify the following URL in your Web browser:

  http://*server_name*/*servlet_location*/HodConfig/trace

  where *server_name* is the name of the zSeries server and *servlet_location*.

  If the configuration servlet is running, the browser window displays trace information from the configuration servlet. This sample trace statement for a doPost request indicates that the servlet is active and is successfully handling requests:

```
Fri Mar 01 13:41:07 EST 2002 (98) Called doPost(/hod)
         [93]: 9.37.3.90 <===> mvs059.raleigh.ibm.com:80 user=null[null]
doPost [93]: got xfer from Pool = null
doPost [93]: null xfer, creating new one ...
doPost [93]: done with create!
doPost [93]: calling doTransfer ...
doTransfer [93]: transfering data to SM ...
Fri Mar 01 13:41:07 EST 2002 (217) [93] POST xfer Client ==> SM 258 bytes.
Fri Mar 01 13:41:07 EST 2002 (596) [93] POST xfer Client <== SM 285 bytes.
doPost [93]: done with transfer!
Fri Mar 01 13:41:07 EST 2002 (634) [93] POST - returning
```

Setting up separate read/write private and publish directories

Set up a separate HFS for the Host On-Demand private directory

When Host On-Demand is installed, files in the /usr/lpp/HOD/hostondemand/private directory are updated in an execution environment, not just by maintenance (PTF) releases. Because this directory is now updated during the Host On-Demand software's execution, it is recommended that you mount a separate (non-service) HFS. You can do this in one of the following ways:

- MOUNT the separate HFS on the current private directory location, /usr/lpp/HOD/hostondemand/private.
- Create a symbolic link to the private directory location as follows:
  1. Do a TSO MKDIR to create a different mount point, such as /etc/HOD/private.
  2. Rename, or back up and delete, your original private directory.

3. Create a symbolic link from the expected location, /usr/lpp/HOD/hostondemand/private, to point to the real location, /etc/HOD/private. Use the following link command:

```
ln -s /etc/HOD/private /usr/lpp/HOD/hostondemand/private
```

Customers running in a sysplex environment using SHARED HFS support can install the Host On-Demand SMP/E managed code in the VERSION HFS, which must be mounted with READ ONLY privileges in a SHARED HFS environment. Make the /private directory a system-specific HFS mounted with READ WRITE privileges, with a symbolic link pointing to the /usr/lpp/HOD/hostondemand/private directory.

If you are using LDAP and native authentication, manually copy the HODrapd and /keys directory to the system-specific /private directory.

When the system-specific /private directory is mounted, it overlays but does not destroy the master /private directory. When maintenance releases are applied, use the master /private directory. If these files were changed, copy them to the system-specific /private directory.

Set up a separate user publish directory

Under Host On-Demand 8, files generated from the Deployment Wizard can be placed in a user-defined directory that is separate from the Host On-Demand publish directory. This makes it easier to apply future Host On-Demand upgrades. It also simplifies installing and maintaining Host On-Demand on OS/390 systems where the SMP/E installed libraries must not contain user modifications (the file systems are mounted read-only). This solution keeps the Host On-Demand publish directory read only and provides a separate writeable location for deploying Deployment Wizard files.

For instructions on deploying Deployment Wizard files in a separate user publish directory and for information on other user-modified files that can be placed outside the publish directory, see “Backing up files and directories” on page 71.

Chapter 17. Configuring Host On-Demand on iSeries

After you install Host On-Demand on the iSeries platform, configure the software as follows:

- To set up the Service Manager, follow the instructions in "Configuring, starting, and stopping the Host On-Demand Service Manager on iSeries".
- To use the Deployment Wizard with an iSeries system, follow the instructions in "Using the Deployment Wizard with iSeries" on page 148.
- To configure security, follow the instructions in "Configuring iSeries servers for secure connection" on page 149.
- To understand the requirements for Unicode support using Coded Character Set Identifiers see "Unicode Support for OS/400" on page 153.

Configuring, starting, and stopping the Host On-Demand Service Manager on iSeries

A menu is provided for starting and stopping the Host On-Demand Service Manager. To access the menu, type the following on the OS/400 command line:

```
GO HOD
```

The following commands can be used from the menu or the OS/400 command line.

Configure (CFGHODSVM)

To configure the Service Manager, choose option 1. You need *JOBCTL and *ALLOBJ authority to use this option. You can configure the following information:

1. Whether to autostart the server when the subsystem starts
2. Adjustment of Java attributes
3. The user ID that the server job uses
4. The subsystem that the server job uses
5. The job description that the server job uses
6. The pre-start class/job priority that the server job uses

There are multiple screens. You may need to page down to see the next screen.

Start (STRHODSVM)

To start the Host On-Demand Service Manager, choose option 2. You need *JOBCTL authority to use this option.

The Service Manager can be automatically started each time that the associated subsystem starts. One way to do this is to add the STRHODSVM command to the system startup program.

To determine whether the Service Manager is running, use the following command:

```
WRKJOB QHODSVM
```

Stop (ENDHODSVM)

To stop the Service Manager, choose option 3. You need *JOBCTL authority to use this option.

Work with HOD Server status

Use this option to view the current status of the Host On-Demand Service Manager.

Certificate Management (WRKHODKYR)

Use this option to work with SSL certificates in one of the Host On-Demand keyrings. Refer to Chapter 5, "Planning for security", on page 35 for general information on SSL related sessions.

Start Information Bundler (STRHODIB)

In the event that you need to contact the IBM Support Center for assistance, use this menu option to gather information about your Host On-Demand configuration.

Create HOD Printer Definition Table (CRTHODPDT)

Use this menu option to create a custom printer definition table for Host On-Demand 3270 printer sessions. A custom printer definition may be necessary if you have a special paper form or if the printer is not supported. Refer to Section 16.5 in the Host Access Client Package Redbook (SG24-6182-00) for additional information.

Start Organizer (STRPCO)

Use this menu option to start the Client Access Organizer for the workstation.

Start a PC Command (STRPCCMD)

Use this menu option to run a command on your PC. You will need to start the Client Access Organizer for the workstation before using this menu option.

Using the Deployment Wizard with iSeries

To use the Deployment Wizard to deploy screens to an iSeries-based Host On-Demand server, do the following:

1. From a Windows workstation, map a network drive to `/qibm` directory on the iSeries system that will be the Host On-Demand server. For additional information, refer to

   `http://publib.boulder.ibm.com/html/`

   `as400/v5r1/ic2924/info/rzahl/rzahlusergoal.htm`
2. Insert the Host On-Demand for Windows CD in the drive. See "Installing the Deployment Wizard" on page 70.
3. A menu will automatically be launched. One of the options is to use the Deployment Wizard. You may run this without having to install the entire Host On-Demand server.
4. Design the custom features and selections.
5. Save the customized HTML file to the mapped network drive (for example, y:\ProdData\hostondemand\hod\myweb).
6. Using a browser, test out the file (for example, http://iSeries.name.com/hod/myweb.html).

Configuring iSeries servers for secure connection

The iSeries servers can be configured to use certificates from a public signing agency or from a private certificate management system, like the AS/400 Digital Certificate Manager. Before you enable SSL, decide which type of certificate to use. See *Deciding where to obtain your digital certificates* on the iSeries Web site (`http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/info/rzain/rzainoverview.htm`)

You must have the following programs installed to use SSL with iSeries:

- Digital Certificate Manager (DCM), option 34 of OS/400
- TCP/IP Connectivity Utilities for AS/400
- IBM HTTP Server for AS/400
- One of the IBM Cryptographic Access Provider products: 40-bit, 56-bit, or 128-bit. The bit size for these products indicates the varying sizes of the digital keys that they employ. A higher bit size results in a more secure connection. Some of these products are not available in all areas due to government export regulations.

Configuring a Telnet server for secure connection

The following table describes the steps to enable Telnet with SSL. You will need to repeat this step for each iSeries system that you wish to use secure connections with.

OS/400 level	Web page (click on the link for more information)
V5R1 and V5R2	*Secure Telnet* on the iSeries Web site (`http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm?info/rzain/rzainrzaintelntpi.htm`) Perform Step 1 only. Client authentication is discussed in "Client authentication" on page 150.
V4R4 and V4R5	*Telnet server and SSL* on the AS/400 Web site (`http://publib.boulder.ibm.com/pubs/html/as400/v4r5/ic2924/info/RZAIWSSLTEL.HTM#HDRRZAIWSSLTEL`)
V4R2 and V4R3	*Telnet SSL Proxy Server* on the AS/400 Web site (`http://www.as400.ibm.com/tstudio/tech_ref/tcp/sslproxy/index.htm`)

Configuring the Host On-Demand CustomizedCAs keyring

If you are using self-signed certificates or certificates from a signing agency that is not in the well-known list, complete the following steps to configure a CustomizedCAs keyring:

1. Type the following command: `GO HOD`.
2. Choose option 5 (Certificate Management).
3. Enter `*CONNECT` for the option and `*CUSTOM` for the name of the keyring, then press the Enter key.
4. Type the TCP/IP name and port for the target server in the following format:
   server.name:*port*

where *server.name* is the TCP/IP name of the target server (for example, my400.myco.com) and *port* is the port for the target server (for example, 992).

This command can take a few minutes to complete. If you are prompted for a password, press the Enter key. If this is the first certificate, a new CustomizedCAs object is created.

5. Select the certificate number that corresponds to the Certificate Authority (CA) that you want to add to the keyring. Be sure to add the CA certificate and not the site certificate. If the port is not responding, refer to "Configuring iSeries servers for secure connection" on page 149.
6. Repeat steps 3-5 for each target server.

To view the contents of the CustomizedCAs keyring, do the following:

1. Type the following command: `GO HOD`.
2. Choose option 5 (Certificate Management).
3. Type `*VIEW` for the option and `*CUSTOM` for the name of the keyring, then press the Enter key.

If you have multiple iSeries machines and would like to create a single certificate that all the machines can use, consider cross certification. Refer to iSeries Wired Security: Protecting Data over the Network, OS/400 Version 5 Release 1DCM and Cryptographic Enhancements (SG24-6168) for additional information about cross certification.

Client authentication

For additional security, consider SSL with client authentication to tightly control who can Telnet to your system over the Internet. For example, you can configure the Telnet server to only allow authentication if the client certificate was issued by your iSeries (through Digital Certificate Manager).

The client certificates have a limited validity period (for example, 90 days). When the certificate expires, the user must perform the Client Certificate Download process in order to continue. This process requires a valid iSeries user ID and password.

Not all Telnet client software is capable of client authentication. When enabled, all SSL-enabled Telnet connections to the iSeries require a user certificate.

OS/400 level	**Detailed instructions (click on the link for more information)**
V5R1 and V5R2	*Secure Telnet* on the iSeries Web site (`http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm?info/rzain/rzainrzaintelntpi.htm`)
V4R4 and V4R5	*Telnet Server; SSL Client Authentication* on the TCP/IP for OS/400 Web site (`http://www.ibm.com/servers/eserver/iseries/tcpip/telnet/ssl.htm`)

Configuring the Host On-Demand OS/400 proxy for secure connections

The OS/400 proxy can be configured to encrypt file transfer and Database On-Demand connections. To do this, the following additional software must be installed on each target iSeries:

- IBM Cryptographic Access Provider
- IBM Client Encryption
- Host Servers
- Digital Certificate Manager

Set up SSL user authorizations

You need to control authorization of the users to the files. To help you to meet the SSL legal responsibilities, you must change the authority of the directory that contains the SSL files to control user access to the files. In order to change the authority, do the following:

1. Enter the command `wrklnk '/QIBM/ProdData/HTTP/Public/jt400/*'`
2. Select option 9 in the directory (SSL40, SSL56, or SSL128).
   a. Ensure *PUBLIC has *EXCLUDE authority.
   b. Give users who need access to the SSL files *RX authority to the directory. You can authorize individual users or groups of users. Remember that users with *ALLOBJ special authority cannot be denied access to the SSL files.

Assign certificates to applications

1. From a web browser, access http://*server.name*:2001 (where *server.name* is the TCP/IP host name of your iSeries system). If you are unable to connect, start the HTTP server with the following OS/400 command:
   ```
   STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
   ```
2. Enter the OS/400 user profile and password (when prompted). You must have *ALLOBJ authority to complete the configuration activities below.
3. Click on Digital Certificate Manager.
4. Click on System Certificates.
5. Click Work with Secure Applications.
6. Click QIBM_OS400_QZBS_SVR_CENTRAL, then click Work with System Certificate.
7. Verify that the *DFTSVR certificate is selected and click Assign New Certificate.
8. Repeat steps 7 and 8 for the following applications:
   - QIBM_OS400_QZBS_SVR_DATABASE
   - QIBM_OS400_QZBS_SVR_DTAQ
   - QIBM_OS400_QZBS_SVR_NETPRT
   - QIBM_OS400_QZBS_SVR_RMTCMD
   - QIBM_OS400_QZBS_SVR_SIGNON
   - QIBM_OS400_QZBS_SVR_FILE
   - QIBM_OS400_QRW_SVR_DDM_DRDA

Repeat the above steps for each target iSeries server.

Configure the OS/400 proxy keyring

If any of the target connections is using self-signed certificates or certificates from a signing agency that is not on the well-known list, do the following:

1. Type the following command: `GO HOD`.

2. Choose option 5 (Certificate Management).
3. Enter `*CONNECT` for the option and `*PROXY` for the name of the keyring, then press the Enter key.
4. Type the TCP/IP name and port for the target server in the following format:
   server.name:*port*

   where *server.name* is the TCP/IP name of the target server (for example, my400.myco.com) and *port* is the port for the sign-on server (for example, 9476).

   This command can take a few minutes to complete. If you are prompted for a password, press the Enter key. If this is the first certificate, a new KeyRing.class object is created.
5. Select the certificate number that corresponds to the Certificate Authority (CA) that you want to add to the keyring.
6. Repeat steps 3-5 for each target server.

Secure Web serving

The Host On-Demand server uses the Web server to download program objects to the browser. This information can be encrypted, but with a considerable performance impact. Refer to the redbook AS/400 HTTP Server Performance and Capacity Planning (SG24-5645) for more information.

The default port for secure web serving is 443. If that port is not enabled, port 80 is used. To enable secure web serving, perform the following steps:

1. From a Web browser, enter: `http://<server.name>:2001` (where <server.name> is the TCP/IP host name of your iSeries). If you are unable to connect, start the HTTP server with the following OS/400 command:
   ```
   STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
   ```
2. Enter the OS/400 user profile and password (when prompted). You must have *ALLOBJ and *SECADM authorities to complete the remaining configuration activities.
3. Click IBM HTTP Server for AS/400.
4. Click Configuration and Administration.
5. Click Configurations.
6. Select the CONFIG configuration from the list.
7. Click Security Configuration.
8. For the Allow HTTP connections and Allow SSL connections selections:
   - Port number (443)
   - Select SSL Client authentication None.
   - Select Apply.
9. Click AS/400 Tasks button on the lower left side of the screen.
10. Click Digital Certificate Manager.
11. Click System Certificates.
12. Click Work with Secure Applications.
13. Click QIBM_HTTP_SERVER_CONFIG; then click Work with System Certificate.
14. Click Assign New Certificate.
15. End the administration HTTP server instance with the following OS/400 command:

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

16. Wait 10 seconds for the HTTP instance to shut down.
17. Start the administration HTTP server instance with the following OS/400 command:

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

18. From a Web browser, enter `https://server.name/hod/hodmain.html` (where *server.name* is the TCP/IP host name of your iSeries).

For more information on a wide variety of iSeries topics, see www.redbooks.ibm.com/tstudio.

Unicode Support for OS/400

General information

In a 5250 Display session, Host On-Demand supports the display of Unicode data located in fields tagged with Coded Character Set Identifiers (CCSIDs). For more information see the following:

- Unicode support for OS/400 using Coded Character Set Identifiers in the online help
- "OS/400 operating systems" on page 13

Host programming information

For host programming information see http://www.ibm.com/eserver/iseries/infocenter.

Chapter 18. Deploying Host On-Demand with WebSphere Portal

As an alternative to accessing Host On-Demand through an HTML file, users can access it through Portal Server, which is a component of WebSphere Portal. Portal Server provides a framework for plugging content extensions known as *portlets* into a Web site. Portlets are applications that run within Portal Server. They organize content from different sources (such as Web sites, e-mail, and business applications) and display it on a single HTML file in a browser window. The WAR or PAR files generated by the Deployment Wizard used to launch Host On-Demand sessions can be deployed as portlets, enabling users to access Host On-Demand through the portal interface. If you are planning to use Host On-Demand and WebSphere Portal Server in conjunction with a firewall, refer to "Using Host On-Demand with a firewall" on page 47.

Both Host On-Demand and Portal Server must be installed to run a Host On-Demand portlet.

How Host On-Demand works with Portal Server

Figure 11 shows how Host On-Demand works with Portal Server.

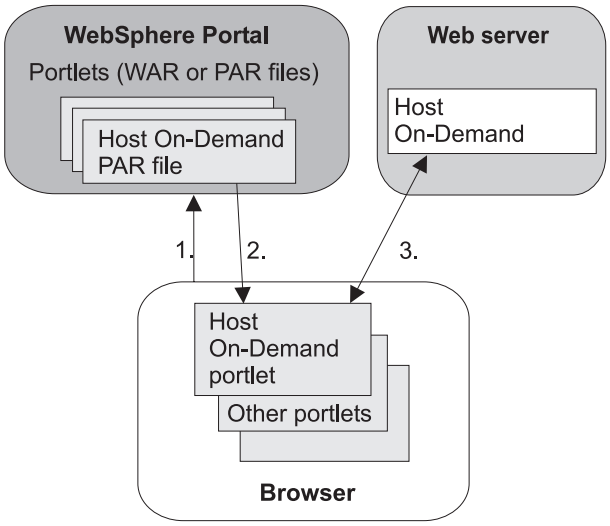

Figure 11. How Host On-Demand works with Portal Server

1. A user logs into the portal through a browser and is authenticated by a user ID and password.
2. The user's customized set of portlets is downloaded to the user's machine and is displayed in the browser.
3. If the user has configured a Host On-Demand portlet, Host On-Demand starts. This gives the user full Host On-Demand functionality within the portlet window, including being able to start sessions and perform other Host On-Demand tasks.

Using Host On-Demand clients with Portal Server

To use Host On-Demand with Portal Server, you need a Host On-Demand portlet. You can quickly and easily create your own custom portlets using the Deployment Wizard. See the Deployment Wizard online help for details about creating portlets. You can also download sample Host On-Demand portlets from the Host On-Demand Service Key site at `http://www6.software.ibm.com/aim/home.html` on the Host On-Demand CSD page under Tools and Utilities.

After you create a custom portlet or obtain a sample one, you can import it directly into Portal Server just like any other portlet. See the WebSphere Portal InfoCenter Web site at `http://www.ibm.com/software/webservers/portal/library.html` for more details.

Limitations on accessing Host On-Demand through a portlet

The Portal supports full Host On-Demand functionality with the following limitations:

- Multiple Host On-Demand portlets cannot be run on the same Portal Server page.
- If the portlet uses caching for Host On-Demand (as configured in the Deployment Wizard), each machine used to access the portlet caches the Host On-Demand client.
- If the portlet configuration allows users to make updates that are saved on their local machines (as specified in the Deployment Wizard), an update made by a user from one machine is not available if that user accesses the portlet from a different machine.
- Host On-Demand bookmarking does not work in the portal environment.
- If the applet size is not configured in the Deployment Wizard, it will default to fixed size, medium.

Special considerations when using a Host On-Demand portlet

When using Host On-Demand with Portal Server, you may want to consider the following issues:

- **Setting the Host On-Demand applet size for the client.** If you would like an applet size that is different from the available options in the Deployment Wizard, you can modify the portlet to specify pixel width and height. To do this, you will first need to extract the portlet and locate the file called WpsHODFinal.jsp. (See the section below titled ″Extending the Host On-Demand Portlets″ for details on extracting and repackaging the portlet.) In this file, locate the two lines beginning with var hod_AppHgt and var hod_AppWid. These are JavaScript variables defining the applet dimensions. Edit the quantities assigned to each of these variables with the dimensions you desire. Save the file, repackage the portlet, and install the portlet in your portal.
- **Host On-Demand sessions when the user logs out of Portal Server.** Host On-Demand runs as an applet on the user's machine and therefore does not know when the user logs out of Portal Server. If the session is running in a separate window (default), the Host On-Demand session will continue until the user either closes the session or closes the browser. If the Host On-Demand session is running embedded in the Portal Server window and the user logs out of Portal Server, the session may appear to have ended, although the connection may remain until the browser window is closed. We strongly recommend that

users close their browser window at the time they log out of Portal Server. In addition, you may wish to configure a session inactivity timeout for your sessions.

- **Session inactivity timeout**. By default, Host On-Demand does not force a timeout on session connections. However, when running a portlet, it may be beneficial to timeout inactive sessions to reduce consumption of resources. The inactivity timeout can be set for most emulator types, including 3270 display and printer sessions, 5250 display and printer sessions, and VT. You can enable and set the timeout parameter Session Inactivity Timeout in minutes for every one of these sessions, except for the 5250 printer, in the `Connection window` of session Properties. For the 5250 printer session, you must set inactivity timeout in the Printer window of session Properties.
- **Installing WebSphere Portal and Host On-Demand on different servers**. If you install WebSphere Portal and Host On-Demand on different servers, certain browsers, such as Netscape 6, may give you a security violation when accessing the Host On-Demand portlet. The problem occurs because some aspects of Host On-Demand functionality rely heavily on the interaction between Java (from the Host On-Demand server) and JavaScript (from WebSphere Portal), and some browsers will not allow the interaction simply because they come from different servers. One solution is to use proxying to make it appear to the browser that WebSphere Portal and Host On-Demand are on the same server. Below is an example of the steps you would need to follow to set up proxying on the Apache/IBM HTTP server:

  1. Configure your Host On-Demand portlet's "HOD Server URL" (hodCodeBase) to point to the host on which WebSphere Portal resides, with the context root of /hod/ (for example, http://portal.company.com/hod).
  2. Uncomment the line (remove the #) in httpd.conf beginning with LoadModule proxy_module.
  3. Add a ProxyPass rule to httpd.conf to convert the HOD Server URL request into a request for the actual Host On-Demand server (for example, ProxyPass /hod/ http://hod.company.com/hod/).
  4. Restart the Web server.

  Now, the client's browser will request Host On-Demand files from the same host as the portal, but these requests will be internally rerouted by the Web server to the actual location of your Host On-Demand install.
- **Caching vs. no caching**. The default setting in the Deployment Wizard is to cache Host On-Demand on each user's machine. Many customers like this option with Host On-Demand because it effectively installs all necessary code on the user's machine and does not require network loads each time the user accesses the HTML file or portlet. However the caching behavior may not be familiar to many Portal Server users, and you may elect to reject the caching option.
- **Choosing the Deployment Wizard model**. The model you choose for your portlet (Configuration-Server, HTML, or Combined) will reflect where your sessions are configured and will determine how user changes are stored. Although Host On-Demand treats portlets the same as HTML files, consider the following characteristics as you decide how to configure your portlet:
  - HTML model: This model has no dependency on the Host On-Demand Configuration Server. If users are allowed to make updates, their changes will be stored on their local machines. These user changes will not be available if the user roams to a different machine.

  - Configuration server-based model: This model requires user access to the Host On-Demand Configuration Server. It allows your users to roam from one machine to another and still see any session modifications they may have made.
  - Combined model: This model requires users to have access to the Host On-Demand Configuration Server in order to obtain the initial session configurations. Any user updates will be saved to the user's local machine and will not be available on a different machine if the user roams.
- **Defining embedded sessions**. By default, Host On-Demand sessions are configured to launch in a separate browser window. You can choose to have the sessions launch in the same window by selecting Preferences > Start Options in Session Properties, and setting Start in a Separate Window to No.
- **Starting the session automatically**. By default, Host On-Demand sessions will not start until the user selects the icon to start. If you wish to have the session start automatically, select Preferences > Start Options in Session Properties and set Start Automatically to Yes.
- **Setting the portlet's access control in Portal Server**. The Host On-Demand portlet does not have any fields that a user can edit using the portlet interface. Therefore, when you import the portlet into Portal Server, you should set the access control to be viewable, but not editable.
- **Specifying unique portlet names in Portal Server**. Use the Page Title field on the File Name and Output Format page in the Deployment Wizard to specify unique portlet names within Portal Server.

Extending the Host On-Demand portlets

Under certain circumstances, you may wish to modify the appearance or functionality of your Host On-Demand portlets. Here are some tips and guidelines to help you extend your portlets:

- Portlet template files are located in the portal subdirectory of your Host On-Demand publish directory (or in your Deployment Wizard installation directory, if you installed it separately). Modifying these templates will affect all portlets that are generated subsequently, so be sure to back up these files if you are going to modify them. Template files include those for the JSPs that are used to display the Host On-Demand applet and those for the XML descriptors that are used to deploy the portlets to WebSphere Portal.
- Each portlet is an archive that can easily be extracted and re-archived using a zip utility or the jar utility packaged with a JRE. Extract the portlet to a temporary directory, preserving directory names. You can then modify the appropriate files, and re-archive the portlet from the top level of the temporary directory.
- XML descriptors are located in the top-level directory of your portlet. JSP files are located in the /PORTLET-INF/hod/html directory for WebSphere Portal Family 2.1, and in the /WEB-INF/hod/html directory for WebSphere Portal 4.1.
- You may wish to add a custom Help file to your portlet. To do this, you must indicate in your portlet.xml file that you support the *help* markup mode. Add a file named WpsHODHelp.jsp (case-sensitive) containing your help information and HTML formatter to your JSP directory in your portlet.
- You may wish to develop a custom portlet that dynamically modifies session properties. Some useful data you may want to access would be the user name of the portal user, or the IP address of the client requesting the page. Consult the portlet APIs on how to access this data. You can use the HTML override syntax

described in Chapter 15, “Modifying session properties dynamically”, on page 123 to then insert data derived from this information into your set of applet parameters.

- Consult the WebSphere Portal InfoCenter installed with WebSphere Portal for detailed information regarding portlet development and APIs.

Chapter 19. Configuring Host On-Demand Server to use LDAP

The Host On-Demand Server is used to manage configuration data for the configuration server-based and combined models. For the default operational mode of the Host On-Demand Server, this data is saved in a non-shared private data store. Some enterprise customers need to manage their configuration information between multiple Host On-Demand servers. If these customers use the non-shared private data store, then their administrators must manage the data for each Host On-Demand Server separately. A Lightweight Directory Access Protocol (LDAP) server directory provides the ability to share user and group configuration information over different instances of the Host On-Demand configuration server.

Using an LDAP directory server to manage and share your definitions across multiple Host On-Demand servers is an option that must be carefully planned and executed. Migration from the private data store, in particular, has implications on the configuration data. LDAP enables the customer to manage the configuration information by arranging users into a hierarchical tree of groups. If existing users are members of more than one group, then some information will be lost. Note that the configuration data in the private data store is not changed when a migration to LDAP occurs. Refer to implications of migrating to LDAP in the Host On-Demand online help for more detailed information.

Setting up LDAP support

1. Decide which LDAP Directory server you are going to use and, if necessary, install it. See “LDAP servers” on page 18 for a list of the LDAP servers supported on your Host On-Demand server platform.
2. If you are running a version of LDAP that does not support the schema for Host On-Demand , install the Host On-Demand schema extension files as described in “Installing the schema extensions” on page 162. (The schema extension files are not required for IBM LDAP Version 3.x or later.)
3. Ask your LDAP administrator for a suffix which Host On-Demand will use to store configuration information. Make a note of the distinguished name (DN) of this suffix; you will need this information to complete the LDAP setup.
4. Ask your LDAP administrator for an administrator DN and password for Host On-Demand; these will be used to authenticate to the LDAP server. The administrator DN must have create, modify and delete privileges for the suffix mentioned in the previous step. Make a note of the DN and password; you will need this information to complete the LDAP setup.
5. Enable LDAP on the Directory Service window in the administration utility. Also, optionally, migrate the private data store configuration information to the LDAP directory server. For more information, refer to Chapter 19, “Configuring Host On-Demand Server to use LDAP”.

Users and groups that are already defined in LDAP for other purposes are not used by Host On-Demand. Users and groups for Host On-Demand must be defined separately by either migrating the configuration information from the private data store or by setting up the users and groups in Host On-Demand after enabling LDAP.

If you are using the IBM LDAP server on Windows and AIX platforms, and you are creating a large number of users, make sure that DB2 is configured with the proper value for APP_CTL_HEAP_SZ. While the value for this variable is dependent on individual installations, setting APP_CTL_HEAP_SZ to 512 is a good starting value.

To configure DB2 heap size in a Windows or AIX environment, issue these commands:

a. set DB2INSTANCE=ldapdb2
b. db2 connect to ldapdb2
c. db2 update db cfg for ldapdb2 using APP_CTL_HEAP_SZ 512
d. db2 force application all
e. db2 terminate
f. db2stop
g. db2start

Also, be sure that STMTHEAP is large enough. The size for these parameters are dependent solely on individual customer configurations and the number of Host On-Demand users that are being migrated to LDAP.

Installing the schema extensions

The Host On-Demand extensions to the LDAP directory schema are provided in several files that are located in the LDAP subdirectory of the publish directory (for example, `your_install_directory\HOD\ldap`, where *your_install_directory* is your Host On-Demand installation directory). These files contain extensions to the LDAP schema and are stored in the standard slapd format. The schema extensions must be in effect before Host On-Demand can store configuration information in an LDAP server. Contact your LDAP administrator to have these schema extensions installed.

Refer to the Program Directory for instructions on installing the schema extensions for the zSeries.

Your LDAP administrator may have already installed these schema extensions for use by another IBM product. If so, skip these steps. If you are using the IBM Directory Server Version 3.1.1 or later, the schema is pre-installed, so you can skip these steps also.

To install the Host On-Demand schema extensions on a Netscape LDAP Directory server:

1. Copy the following slapd files from the <Host On-Demand publish directory>/ldap directory to the Netscape LDAP config directory on the LDAP server :

   ```
   Netscape.IBM.at
   Netscape.IBM.oc
   ```

2. Stop the LDAP server.
3. Edit the <Netscape LDAP config directory>/`slapd.conf` file and add the following statements:

   ```
   userat "<Netscape LDAP config directory>/Netscape.IBM.at"
   useroc "<Netscape LDAP config directory>/Netscape.IBM.oc"
   ```

4. Restart the LDAP Server.

To install the Host On-Demand schema extensions on an IBM LDAP Directory server:

1. Copy the following slapd files from the Host On-Demand publish directory/ldap directory to the <installation directory>/etc directory on your LDAP server:

   ```
   V2.1.IBM.at
   V2.1.IBM.oc
   ```

2. Stop the LDAP server.
3. Edit the <installation directory>/etc/`slapd.at.conf` file and add the following statement to the end of the file:

   ```
   include /etc/V2.1.IBM.at
   ```

4. Edit the <installation directory>/etc/`slapd.oc.conf` file and add the following statement to the end of the file:

   ```
   include /etc/V2.1.IBM.oc
   ```

5. Restart the LDAP server.

Configuring the Host On-Demand server to use LDAP as a data store

1. Open the Administration window and logon to Host On-Demand.
2. Click Services > Directory Service
3. Click the Use Directory Service (LDAP) box and then enter the LDAP server information.

   Destination Address
   Type the IP address of the LDAP directory. Use either the host name or dotted decimal format. The default is the host name of the Host On-Demand server.

   Destination Port
   Type the TCP/IP port on which the LDAP server will accept a connection from an LDAP client. The default port is 389.

   Administrator Distinguished Name
   Type the distinguished name (DN) of the directory administrator that allows Host On-Demand to update information. You must use the LDAP string representation for distinguished names (for example, `cn=Chris Smith,o=IBM,c=US`).

   Administrator Password
   Type the directory administrator's password.

   Distinguished Name Suffix
   Type the distinguished name (DN) of the highest entry in the directory information tree (DIT) for which information will be saved. Host On-Demand will store all of its configuration information below this suffix in the DIT. You must use the LDAP string representation for distinguished names (for example, `cn=HOD,o=IBM,c=US`).

   Migrate Configuration to Directory Service
   To migrate users and groups from the private data store to the LDAP directory, click the check box. Migrating to LDAP has significant implications for your group and user configuration information. Refer to LDAP Migration Implications in the online help for more information. You can check this box either when you switch to the directory server, or after you have made the switch.

The Redirector configuration is not migrated to the directory server.

If you have a problem connecting to LDAP and migrating, try to connect to LDAP first. Then, after successfully connecting, try to migrate.

4. Click Apply.

   When you are asked to authenticate with the LDAP directory for the first time, specify a user ID of "admin" and a password of "password". You can change this password after the first log on. Even though you might have changed your password for the private data store, that ID and password continues to be valid for the private data store only. For the LDAP directory, a separate user ID and password are required. To avoid confusion, you can change your LDAP directory password to be the same as your private data store password.

Changes made on this panel are effective immediately. Once you have switched to the LDAP server, subsequent user-related changes will be made only on the LDAP server, including administrative changes to groups, users, or sessions, and changes such as new passwords, macros, keyboard changes, etc., by either the administrator or a user.

Appendix A. Using locally installed clients

The locally installed client installs to a local disk. The client applet is loaded directly into the default system browser, so there is no download from a server. The most common reason to configure a local client is for users who connect remotely over slow telephone lines, where download time can be an issue and connectivity is unpredictable. You can also use the locally installed client to test host access capabilities without installing the full Host On-Demand product.

Operating systems that support the locally installed client

Host On-Demand can be installed as a client on the following operating systems:

- Windows 98
- Windows NT 4.0 with SP5 or later
- Windows Millennium (Me)
- Windows 2000
- Windows XP (32-bit)

The locally-installed client requires approximately 320 MB of disk space.

Installing the local client

To install the Host On-Demand local client on a Windows NT, Windows 2000, or Windows XP workstation, you must be a member of the Administrators group.

1. Insert the CD and run `hodinstallwin.exe -lc` from the `\hodinstall` directory of the CD.
2. Click Install.
3. Proceed through the rest of the windows.
4. If you have not already done so, read the Readme available in the last window.

At the end of installation, the Host On-Demand Service Manager is configured and started automatically. On Windows NT, Windows 2000, and Windows XP, the Service Manager is installed as a Service; on Windows 98, and Windows Millennium (Me) it is added to the Start menu.

Starting the local client

To start Host On-Demand as a client, click Start > Programs > IBM WebSphere Host On-Demand > Host On-Demand.

Removing the local client

To remove the local client, use Add/Remove Programs from the Control Panel. If InstallShield does not remove the `hostondemand` directory, you must remove it manually.

Appendix B. Using the IKEYCMD command-line interface

IKEYCMD is a command-line tool, in addition to the Host On-Demand Certificate Management Utility, that can be used to manage keys, certificates, and certificate requests. It is functionally similar to Certificate Management and is meant to be run from the command line without a graphical interface. It can be called from native shell scripts and programs to be used when applications prefer to add custom interfaces to certificate and key management tasks. It can create key database files for all of the types that the Certificate Management utility currently supports. It can create certificate requests, import CA-signed certificates and manage self-signed certificates. It is Java-based and is available only on Windows and AIX platforms.

Use IKEYCMD for configuration tasks related to public-private key creation and management. You cannot use IKEYCMD for configuration options that update the server configuration file, httpd.conf. For options that update the server configuration file, you must use the IBM Administration Server.

Environment set-up for IKEYCMD command-line interface

Set up the environment variables to use the IKEYCMD command-line interface as follows:

For Windows platforms, do the following:

- Using the user interface or by modifying autoexec.bat on a command window, set/modify the PATH variable to include the location of the Java executable files:

  ```
  set PATH=c\Program Files\IBM\HostOnDemand\bin;%PATH%;
  ```

- Using the user interface or by modifying autoexec.bat on a command window, set/modify the CLASSPATH environment variable as follows:

  ```
  set CLASSPATH=c\Program Files\IBM\GSK6\classes\cfwk.zip;C:\
  Program Files\IBM\GSK6\classes\gsk6cls.jar;%CLASSPATH%;
  ```

For AIX platforms:

First ensure that your xlC files (which constitute the run-time library for the standard AIX C++ compiler) meet one of the following requirements:

- on AIX 4.3: fileset xlC.aix43.rte must be at level 5.0.2.0 or later
- on AIX 5.2: fileset xlC.aix50.rte must be at level 6.0.0.3 or later

Use the following command to confirm your version:

```
lslpp -ha "xlC.aix*.rte"
```

(If your xlC fileset is outdated and you start the Host On-Demand ServiceManager with Certificate Management active, errors occur.)

Next make the following specifications:

- Set your PATH to where your Java or JRE executable resides:

  ```
  EXPORT PATH=/opt/IBM/HostOnDemand/jre:$PATH
  ```

- Set the following CLASSPATH environment variable:

  ```
  EXPORT CLASSPATH=/usr/opt/ibm/gskak/classes/cfwk.zip:/
  usr/opt/ibm/gskak/classes/gsk6cls.jar:$CLASSPATH
  ```

Once you have completed these steps, IKEYCMD should run from any directory. To run an IKEYCMD command, use the following syntax:

```
java com.ibm.gsk.ikeyman.ikeycmd <command>
```

IKEYCMD command-line syntax

The syntax of the Java CLI is

```
java [-Dikeycmd.properties=<properties_file>],
com.ibm.gsk.ikeyman.ikeycmd <object> <action> [options]
```

where

- -Dikeycmd.properties specifies the name of an optional properties file to use for this Java invocation. A default properties file, ikminit_hod.properties, is provided as a sample file that contains the default settings for Host On-Demand.
- Object is one of the following:
  - -keydb: actions taken on the key database (either a CMS key database file or SSLight class)
  - -cert: actions taken on a certificate
  - -certreq: actions taken on a certificate request
  - -help: display help for the IKEYCMD invocations
  - -version: display version information for IKEYCMD

Action is the specific action to be taken on the object, and options are the options, both required and optional, specified for the object and action pair.

The object and action keywords are positional and must be specified in the selected order. However, options are not positional and can be specified in any order, provided that they are specified as an option and operand pair.

IKEYCMD list of tasks for Host On-Demand

IKEYCMD command-line interface tasks required for Host On-Demand are summarized in the following sections of this appendix:

Creating a new key database

A key database is a file that the server uses to store one or more key pairs and certificates. This is required to enable secure connections between the Host On-Demand server and clients. Before configuring SSL communication, you must create the HODServerKeyDb.kdb key database file in *your_install_directory*\bin for

Windows and *your_install_directory*/bin for AIX. This file is not shipped with Host On-Demand, so you must create it after the first install.

For Windows platforms, for example, to create a new key database using the IKEYCMD command-line interface, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -keydb -create
-db your_install_directory\bin\HODServerKeyDb.kdb
-pw <password> -type cms -expire <days> -stash
```

where *your_install_directory* is your Host On-Demand installation directory.

Note the following descriptions:

- <password>: Password is required for each key database operation. Even though a database of the type sslight requires a specified password, the password can be a NULL string (specified as "").
- -type: the HODServerKeyDb.kdb used by the Host On-Demand server is of the type CMS.
- -expire: Days before password expires.
- -stash: Stashes password for key database. Stashing the password is required for the IBM HTTP Server and the Host On-Demand server.

  When the -stash option is specified during the key database creation, the password is stashed in a file with the filename HODServerKeyDb.sth

  Once the HODServerKeyDb.kdb file has been created, it holds all the security information needed by the Host On-Demand server. Any additions or changes are made to the existing HODServerKeyDb.kdb key database file.

Whenever you create or make changes to the HODServerKeyDb.kdb file, you must stop and restart the Host On-Demand Service Manager.

Setting the database password

When you create a new key database, you specify a key database password. This password protects the private key. The private key is the only key that can sign documents or decrypt messages encrypted with the public key. Changing the key database password frequently is a good practice.

Use the following guidelines when specifying the password:

- The password must be from the U.S. English character set.
- The password should be at least six characters and contain at least two nonconsecutive numbers. Make sure the password does not consist of publicly obtainable information about you, such as the initials and birth date for you, your spouse, or children.
- Stash the password.

Keep track of expiration dates for the password. If the password expires, a message is written to the error log. The server will start, but there will not be a secure network connection if the password has expired.

Changing the database password

To change the database password, do the following:

For Windows platforms, for example, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -keydb -changepw
-db your_install_directory\bin\HODServerKeyDb.kdb
-pw <password> -new_pw <new_password> -expire <days> -stash
```

where *your_install_directory* is your Host On-Demand installation directory.

Note the following descriptions:

- -new_pw: New key database password; this password must be different than the old password, and this password cannot be a NULL string.
- -expire: Days before password expires.
- -stash: Stashes password for key database. Stashing the password is required for the IBM HTTP Server and the Host On-Demand server.

Listing CAs

To display a list of trusted CAs in the HODServerKeyDb.kdb key database, do the following:

For Windows platforms, for example, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -list CA
-db your_install_directory\bin\HODServerKeyDb.kdb
-pw <password>-type cms
```

where *your_install_directory* is your Host On-Demand installation directory.

By default, HODServerKeyDb.kdb comes with the CA certificates of the following well-known trusted CAs:

- IBM World Registry CA
- Integrion CA Root (from IBM World Registry)
- VeriSign Class 1 Public Primary CA
- VeriSign Class 2 Public Primary CA
- VeriSign Class 3 Public Primary CA
- VeriSign Class 4 Public Primary CA
- VeriSign Test CA
- RSA Secure Server CA (from VeriSign)
- Thawte Personal Basic CA
- Thawte Personal Freemail CA
- Thawte Personal Premium CA
- Thawte Premium Server CA
- Thawte Server CA

Creating a new key pair and certificate request

To create a public-private key pair and certificate request, do the following:

1. For Windows platforms, for example, enter the following command:

   ```
   java com.ibm.gsk.ikeyman.ikeycmd -certreq -create
   -db your_install_directory\bin\HODServerKeyDb.kdb
   -pw <password> -size <1024 | 512> -dn <distinguished_name>
   -file <filename> -label <label>
   ```

where *your_install_directory* is your Host On-Demand installation directory.

Note the following descriptions:

- -size: key size of 512 or 1024
- -label: label attached to certificate or certificate request
- -dn: X.500 distinguished name. This is input as a quoted string of the following format: (Only CN, O, and C are required; CN=common_name, O=organization, OU=organization_unit, L=location, ST=state/province, C=country.)

  ```
  "CN=weblinux.raleigh.ibm.com,O=ibm,OU=IBM HTTP Server,L=RTP,ST=NC,C=US"
  ```

- -file: name of file where the certificate request will be stored. By default, Host On-Demand uses the name certreq.arm and it should be stored in *your_install_directory*\bin (where *your_install_directory* is your Host On-Demand installation directory), where HODServerKeyDb.kdb is located.

2. Verify that the certificate was successfully created.

   a. View the contents of the certificate request file you created.

   b. Make sure the key database recorded the certificate request:

   ```
   java com.ibm.gsk.ikeyman.ikeycmd -certreq -list
   -db <filename> -pw <password>
   ```

   You should see the label listed that you just created.

3. Send the newly created file to a certificate authority.

Storing the server certificate

Receiving a CA-signed certificate

Use this procedure to receive an electronically mailed certificate from a certificate authority (CA), designated as a trusted CA on your server. By default, the following CA certificates are stored in the HODServerKeyDb.kdb key database and marked as trusted CA certificates:

- IBM World Registry CA
- Integrion CA Root (from IBM World Registry)
- VeriSign Class 1 Public Primary CA
- VeriSign Class 2 Public Primary CA
- VeriSign Class 3 Public Primary CA
- VeriSign Class 4 Public Primary CA
- VeriSign Test CA
- RSA Secure Server CA (from VeriSign)
- Thawte Personal Basic CA
- Thawte Personal Freemail CA
- Thawte Personal Premium CA
- Thawte Premium Server CA
- Thawte Server CA

The Certificate Authority may send more than one certificate. In addition to the certificate for your server, the CA may also send additional Signing certificates or Intermediate CA Certificates. For example, Verisign includes an Intermediate CA Certificate when sending a Global Server ID certificate. Before receiving the server

certificate, receive any additional Intermediate CA certificates. Follow the instructions in “Storing a CA certificate” to receive Intermediate CA Certificates.

If the CA who issues your CA-signed certificate is not a trusted CA in the key database, you must first store the CA certificate and designate the CA as a trusted CA. Then you can receive your CA-signed certificate into the database. You cannot receive a CA-signed certificate from a CA who is not a trusted CA. For instructions, see “Storing a CA certificate”

For Windows platforms, for example, to receive the CA-signed certificate into a key database, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -receive -file <filename>
-db your_install_directory\bin\HODServerKeyDb.kdb -pw <password>
-format <ascii | binary> -default_cert <yes | no>
```

where *your_install_directory* is your Host On-Demand installation directory.

Note the following descriptions:

- -format: Certificate Authority might provide CA Certificate in either ASCII or binary format
- -label: Label attached to CA certificate.
- -trust: Indicates whether this CA can be trusted. Use enable options when receiving a CA certificate.
- -file: File containing the CA certificate.

Storing a CA certificate

For Windows platforms, for example, to store a certificate from a CA who is not a trusted CA, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -add
-db your_install_directory\bin\HODServerKeyDb.kdb
-pw <password> -label <label> -format <ascii | binary>
-trust <enable |disable> -file <file>
```

where *your_install_directory* is your Host On-Demand installation directory.

Note the following descriptions:

- -label: Label attached to certificate or certificate request
- -format: Certificate Authorities might supply a binary ASCII file
- -trust: Indicate whether this CA can be trusted. This should be Yes.

You must stop and restart the Host On-Demand Service Manager after doing this.

Creating a self-signed certificate

It usually takes two to three weeks to get a certificate from a well-known CA. While waiting for an issued certificate, use IKEYCMD to create a self-signed server certificate to enable SSL sessions between clients and the server. Use this procedure if you are acting as your own CA for a private Web network.

For Windows platforms, for example, to create a self-signed certificate, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -create
-db your_install_directory\bin\HODServerKeyDb.kdb
-pw <password> -size <1024 | 512> -dn <distinguished name>
-label <label> -default_cert <yes or no>
```

where *your_install_directory* is your Host On-Demand installation directory.

Note the following descriptions:

- -size: Key size 512 or 1024
- -label: Enter a descriptive comment used to identify the key and certificate in the database.
- -dn: Enter an X.500 distinguished name. This is input as a quoted string of the following format (Only CN, O, and C are required; CN=common_name, O=organization, OU=organization_unit,L=location, ST=state, province, C=country).

  ```
  "CN=weblinux.raleigh.ibm.com,O=ibm,OU=IBM HTTP Server,L=RTP,ST=NC,C=US"
  ```
- -default_cert: Enter yes, if you want this certificate to be the default certificate in the key database. If not, enter No.

Making server certificates available to clients

All the certificates in the HODServerKeyDb.kdb are available to the Host On-Demand server. However, in some of the configurations, one of these certificates must also be made available to the clients that access the server. In the cases where your server uses a certificate from an unknown CA, the root of that certificate must be made available to the client. If your server uses a self-signed certificate, then a copy of that certificate must be made available to the clients.

For Host On-Demand downloaded and cached clients, this is done by extracting the certificate to a temporary file and creating or updating a file named CustomizedCAs.p12, which should be present in the Host On-Demand publish directory.

To create the CustomizedCAs.p12 file for downloaded or cached clients, enter the following command:

```
java com.ibm.gsk.ikeyman -keydb -create -db
CustomizedCAs.p12 -pw hod -type pkcs12
```

The default password is hod.

For older clients, you need to create and update the CustomizedCAs.class file for download or cached clients by entering the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -keydb
-create -db CustomizedCAs.class -type sslight
```

It will prompt for a password. Simply press Enter, which implies a NULL password. After the CustomizedCAs.class file has been created, you will need to add the server certificate to it.

Adding the root of an unknown CA to CustomizedCAs.p12

First, extract the CA's root certificate or a self-signed certificate from the HODServerKeyDb.kdb key database file. To do this for Windows, for example, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -extract
-db your_install_directory\bin\HODServerKeyDb.kdb
-pw <password> -label <label> -target cert.arm -format ascii
```

where *your_install_directory* is your Host On-Demand installation directory.

Note the following descriptions:

- -label : Label attached to the certificate.
- -pw: password to open HODServerKeyDb.kdb key database file.
- -target : Destination file or database. In this case, it is the name of the Base-64 Armored ASCII format file with a default filename of cert.arm.
- -format: Can be either ASCII or Binary.

Now, add this CA root certificate to the CustomizedCAs.p12 file. To add a CA root certificate or a self-signed certificate to the list of signers in CustomizedCAs.p12, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -add
-db CustomizedCAs.p12 -pw hod -label <label>
-file cert.arm -format ascii -trust <enable | disable>
```

For older clients, to add this CA root certificate to the CustomizedCAs.class file, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -add
-db CustomizedCAs.class -label <label>
-file cert.arm -format ascii -trust <enable | disable>
```

Note the following descriptions:

- -label: Label for the certificate being added.
- -file: Name of the file where the certificate has been extracted to. In this case, it is the name of the Base-64 Armored ASCII format file with a default filename of cert.arm.
- -format: Can be ASCII or Binary.
- -trust: Decides whether to set as a trusted root. Enable will set the CA root or self-signed certificate as a trusted root. Disable will not set the CA root or self-signed certificate as a trusted root.

Stop and restart the Host On-Demand Service Manager after completing this task.

Exporting keys

To export keys to another key database or to export keys to a PKCS12 file, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -export -db <filename>
-pw <password> -label <label> -type <cms | sslight>
-target <filename> -target_pw <password>
-target_type <cms | sslight | pkcs12> -encryption <strong | weak>
```

Note the following descriptions:

- -label : Label attached to the certificate.
- -target : Destination file or database.

- -target_pw : Password for the target key database.
- -target_type : Type of the database specified by -target operand
- -encryption : Strength of encryption. Default is strong.

Importing keys

To import keys from another key database, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -import -db <filename>
-pw <password> -label <label> -type <cms | sslight> -target
<filename> -target_pw <password> -target_type <cms | sslight>
```

To import keys from a PKCS12 file,enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -import -file <filename>
-pw <password> -type pkcs12 -target <filename>
-target_pw <password> -target_type <cms | sslight>
```

Note the following descriptions:

- -label: Label attached to the certificate.
- -target: Destination database.
- -target_pw: Password for the key database if -target specifies a key database
- -target_type : Type of the database specified by -target operand.

Showing the default key in a key database

For Windows platforms, for example, to display the default key entry, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -getdefault
-db your_install_directory\bin\HODServerKeyDb.kdb
-pw <password>
```

where *your_install_directory* is your Host On-Demand installation directory.

Storing the encrypted database in a stash file

For a secure network connection, store the encrypted database password in a stash file. For Windows platforms, for example, to store the password while a database is created, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -keydb -create
-db your_install_directory\bin\HODServerKeyDb.kdb
-pw <password> -type cms -expire <days> -stash
```

where *your_install_directory* is your Host On-Demand installation directory.

For Windows platforms, for example, to store the password after a database has been created, enter the following command:

```
java com.ibm.gsk.ikeyman.ikeycmd -keydb -stashpw
-db your_install_directory\bin\HODServerKeyDb.kdb -pw <password>
```

where *your_install_directory* is your Host On-Demand installation directory.

Using GSK5CMD batch file

A batch file, gsk5cmd, provides the same function of the ″java com.ibm.gsk.ikeyman″ command. For Windows platforms, for example, to store the password after a database has been created, you can also enter following command:

```
gsk5cmd -keydb -stashpw
-db your_install_directory\bin\HODServerKeyDb.kdb -pw <password>
```

where *your_install_directory* is your Host On-Demand installation directory.

IKEYCMD command-line parameter overview

The following table describes each action that can be performed on a specified object.

Object	Action	Description
-keydb	-changepw	Change the password for a key database
	-convert	Convert the key database from one format to another
	-create	Create a key database
	-delete	Delete the key database
	-stashpw	Stash the password of a key database into a file
-cert	-add	Add a CA certificate from a file into a key database
	-create	Create a self-signed certificate
	-delete	Delete a CA certificate
	details	List the detailed information for a specific certificate
	-export	Export a personal certificate and its associated private key from a key database into a PKCS#12 file, or to another key database
	-extract	Extract a certificate from a key database
	-getdefault	Get the default personal certificate
	-import	Import a certificate from a key database or PKCS#12 file
	-list	List all certificates
	-modify	Modify a certificate (NOTE: Currently, the only field that can be modified is the Certificate Trust field)
	-receive	Receive a certificate from a file into a key database

	-setdefault	Set the default personal certificate
	-sign	Sign a certificate stored in a file with a certificate stored in a key database and store the resulting signed certificate in a file
-certreg	-create	Create a certificate request
	-delete	Delete a certificate request from a certificate request database
	-details	List the detailed information of a specific certificate request
	extract	Extract a certificate request from a certificate request database into a file
	-list	List all certificate requests in the certificate request database
	-recreate	Recreate a certificate request
-help		Display help information for the IKEYCMD command
-version		Display IKEYCMD version information

IKEYCMD command-line options overview

The following table shows each option that can be present on the command line. The options are listed as a complete group; however, their use is dependent on the object and action specified on the command line.

Option	Description
-db	Fully qualified path name of a key database
-default_cert	Sets a certificate to be used as the default certificate for client authentication (yes or no). The default is no.
-dn	X.500 distinguished name. Input as a quoted string of the following format (only CN, O, and C are required): ″CN=Jane Doe,O=IBM,OU=Java Development,L=Endicott, ST=NY,ZIP=13760,C=country″
-encryption	Strength of encryption used in certificate export command (strong or weak). The default is strong.
-expire	Expiration time of either a certificate or a database password (in days). Defaults are 365 days for a certificate and 60 days for a database password.

-file	File name of a certificate or certificate request (depending on specified object)
-format	Format of a certificate (either ascii for Base64_encoded ASCII or binary for Binary DER data). The default is ascii.
-label	Label attached to a certificate or certificate request
-new_format	New format of key database
-new_pw	New database password
-old_format	Old format of key database
-pw	Password for the key database or PKCS#12 file. See "Creating a new key database" on page 168.
-size	Key size (512 or 1024). The default is 1024.
-stash	Indicator to stash the key database password to a file. If specified, the password will be stashed in a file.
-target	Destination file or database.
-target_pw	Password for the key database if -target specifies a key database. See "Creating a new key database" on page 168.
-target_type	Type of database specified by -target operand (see -type).
-trust	Trust status of a CA certificate (enable or disable). The default is enable.
-type	Type of database. Allowable values are cms (indicates a CMS key database), webdb (indicates a keyring), sslight (indicates an sslight .class), or pkcs12 (indicates a PKCS#12 file).
-x509version	Version of X.509 certificate to create (1, 2 or 3). The default is 3.

Command-line invocation

The following is a list of each of the command line-invocations, with the optional parameters specified in italics.

For simplicity, the actual Java invocation, java com.ibm.gsk.ikeyman.ikeycmd, is omitted from each of the command invocations.

```
-keydb -changepw -db <filename> -pw <password>
-new_pw <new_password> -stash -expire <days>

-keydb -convert -db <filename> -pw <password>
-old_format <cms | webdb> -new_format <cms>

-keydb -create -db <filename> -pw <password> -type <cms | sslight>
-expire <days> -stash

-keydb -delete -db <filename> -pw <password>

-keydb -stashpw -db <filename> -pw <password>

-cert -add -db <filename> -pw <password> -label <label>
-file <filename> -format <ascii | binary> -trust <enable | disable>
```

```
-cert -create -db <filename> -pw <password> -label <label>
-dn <distinguished_name> -size <1024 | 512> -x509version <3 | 1 | 2>
-default_cert <no | yes>

-cert -delete -db <filename> -pw <password> -label <label>

-cert -details -db <filename> -pw <password> -label <label>

-cert -export -db <filename> -pw <password> -label <label>
-type <cms | sslight> -target <filename> -target_pw <password>
-target_type <cms | sslight | pkcs12> -encryption <strong | weak>

-cert -extract -db <filename> -pw <password> -label <label>
-target <filename> -format <ascii | binary>

-cert -getdefault -db <filename> -pw <password>

-cert -import -db <filename> -pw <password> -label <label>
-type <cms | sslight> -target <filename> -target_pw <password>
-target_type <cms | sslight>

-cert -import -file <filename> -type <pkcs12> -target <filename>
-target_pw <password> -target_type <cms | sslight>

-cert -list <all | personal | CA | site> -db <filename>
-pw <password> -type <cms | sslight>

-cert -modify -db <filename> -pw <password> -label <label>
-trust <enable | disable>

-cert -receive -file <filename> -db <filename> -pw <password>
-format <ascii | binary> -default _cert <no | yes>

-cert -setdefault -db <filename> -pw <password> -label <label>

-cert -sign -file <filename> -db <filename> -pw <password>
-label <label> -target <filename> -format <ascii | binary>
-expire <days>

-certreq -create -db <filename> -pw <password> -label <label>
-dn <distinguished_name> -size <1024 | 512> -file <filename>

-certreq -delete -db <filename> -pw <password> -label <label>

-certreq -details -db <filename> -pw <password> -label <label>

-certreq -extract -db <filename> -pw <password> -label <label>
-target <filename>

-certreq -list -db <filename> -pw <password>

-certreq -recreate -db <filename> -pw <password> -label <label>
-target <filename>

-help

-version
```

User properties file

In order to eliminate some of the typing on the Java CLI invocations, user properties can be specified in a properties file. The properties file can be specified on the Java command-line invocation via the -Dikeycmd.properties Java option. For Windows platforms, a sample properties file, ikminit_hod.properties, is supplied in *your_install_directory*\bin, where *your_install_directory* is your Host On-Demand installation directory. For AIX platforms, this file is supplied in *your_install_directory*/bin. These installation directories contain the default setting for Host On-Demand.

Appendix C. P12 Keyring utility

A graphical Certificate Management utility (available on Windows and AIX platforms) is provided to allow you to create certificate requests, receive and store certificates, and create self-signed certificates. The P12 Keyring utility is provided mainly for platforms that do not have the Certificate Management Utility to create a keyring database with root certificates of self-signed and unknown Certificate Authority certificates. However, it can be used on any Host On-Demand platform. This utility provides system administrators with an easy way to create and deploy an SSL keyring database.

The P12 Keyring utility is written in Java. It obtains a server certificate from a Telnet or an FTP server (or a Redirector) that is configured for SSL. An SSL connection is made to the specified server and SSL port. If the port is not provided, the well-known secure Telnet or FTP port is used. The server's certificate will be extracted and added to the specified p12 file.

Access to the keyring database is password-protected. A password prompt will be given before any of the commands are performed. If the specified keyring file does not exist, it will be created and the password will be stored in the file.

The Host On-Demand SSL support requires the password to be `hod`. If you are adding a private certificate to the keyring database, another password prompt will be given for the second p12 file.

Usage

```
P12Keyring p12FileName connect ipaddr[:port] [ftp]
P12Keyring p12FileName add p12FileName2
P12Keyring p12FileName list
```

Options

connect - establishes an SSL connection to the specified ipaddr and port. The port number and ftp keyword are optional. If the port number is not specified, the default secure Telnet port 443 or the default secure FTP port 990 will be used.

If the **ftp** keyword is specified, the connection is to be made to a secure FTP server that is configured for security. There are two types of security options for FTP servers:

- Implicit security to port 990
- Explicit security to any other port

If the ftp keyword is specified but the port number is not specified or it is 990, implicit security negotiations are performed. If the ftp keyword is specified and the port number is not 990, explicit security negotiations are done by issuing `AUTH` command first.

add - adds a private client certificate to the specified keyring database.

list - displays a list of certificates stored in the specified keyring database.

Examples

Windows:

```
C:\your_install_dir\lib\P12Keyring c:\your_install_dir\HOD\CustomizedCAs
connect myServer.raleigh.ibm.com:702

C:\your_install_dir\lib\P12Keyring c:\your_install_dir\HOD\CustomizedCAs
connect myFTPServer.raleigh.ibm.com:5031 ftp
```

where *your_install_dir* is your Host On-Demand installation directory.

Unix:

```
Java –classpath .;cd your_install_dir/lib/sm.zip \
com.ibm.hod5sslight.tools.P12Keyring CustomizedCAs connect
myServer.raleigh.ibm.com:702
```

where *your_install_dir* is your Host On-Demand installation directory.

Appendix D. Native platform launcher command line options

When you enter the following command line options with your native platform launcher, the launcher passes them to the Host On-Demand install as installation parameters. Options that suppress the GUI wizard are marked accordingly.

Table 20. Command line options

Option	Purpose	Example usage
`-console` (Suppresses the GUI wizard)	Installs Host On-Demand in console mode.	`hodinstallwin.exe -console`
`-log #!`*filename* where # echoes the display to standard output and !*filename* is the name of the log file. If you specify ! without a file name, the default log file name is used.	Generates an installation file log with the name specified.	`hodinstallwin.exe -log #!\mydirectory\logfile`
`-options` *filename*	Installs Host On-Demand with command line options that set specified properties for the installation.	`hodinstallwin.exe -silent -options c:\mydirectory\responseFile`
`-options-record` *filename*	Generates an options text file recording your responses to the Host On-Demand install wizard, establishing them as default values for installation variables.	`hodinstallwin.exe -options-record responses.txt`
`-options-template` *filename*	Generates an options text file containing the default installation values.	`hodinstallwin.exe -options-template template.txt`
`-silent` (Suppresses the GUI wizard)	Installs Host On-Demand in silent mode, accepting all default installation values.	`hodinstallwin.exe -silent`

The following additional command line options apply only to the *process* of calling and running the installation program. Enter them at the command line with the native platform launcher.

Table 21. Launch-specific command line options

Option	Purpose	Example usage
`-is:log`*filename*	Generates a log file for the native launcher's JVM searches.	`hodinstallwin.exe -is:log myLogFile.txt`
`-is:silent`	Prevents the display of the launcher user interface (UI) while JVM searches and other initializations are taking place. (Commonly used with the command line option `silent`.)	`hodinstallwin.exe -is:silent`

Table 21. Launch-specific command line options (continued)

Option	Purpose	Example usage
`-is:tempdir`*directory*	Sets the temporary directory used by the Host On-Demand install.	`hodinstallwin.exe -is:tempdir "c:\temp"`

Appendix E. 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 F. 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 U.S.A.

SC31-6301-02