



IBM WebSphere  
Host Publisher  
Version  
3.5  
Implementation Guide

Host Publisher Version 3.5  
White Paper

Introduction .....	5 of 42
<b>New Features Overview</b> .....	5 of 42
Host Publisher Prerequisites .....	6 of 42
<b>Supported versions of WebSphere</b>	
<b>Application Server</b> .....	6 of 42
<b>Installing the Web server</b> .....	7 of 42
Host Publisher Application	
Migration from Host Publisher	
2.x to 3.5 Information .....	7 of 42
<b>Migrating JavaServer Pages (JSPs)</b> .....	7 of 42
<b>WebSphere Studio 3.5 and Host</b>	
<b>Publisher 2.1 Integration Objects</b> .....	8 of 42
Migration from Host Publisher	
2.2 and WebSphere 3.0.2.1 to	
Host Publisher 3.5 and	
WebSphere 3.5.3 .....	8 of 42
<b>What you must know before installing</b>	
<b>Host Publisher version 3.5</b> .....	8 of 42
<b>Hardware and Software Requirements</b> .....	9 of 42
<b>Limitations</b> .....	11 of 42
Upgrading from a Previous	
version of Host Publisher .....	11 of 42
<b>Testing installation of the Host</b>	
<b>Publisher Server</b> .....	13 of 42
Host Publisher Distributed	
Administration .....	15 of 42
<b>Accessing Host Publisher Server</b>	
<b>Administration</b> .....	15 of 42
<b>Securing access to Host Publisher</b>	
<b>Server Administration using WebSphere</b>	
<b>Application Server</b> .....	17 of 42
JVM cloning and load	
balancing in WebSphere .....	18 of 42
<b>Load balancing of cloned Web</b>	
<b>applications</b> .....	18 of 42
<b>Load balancing of cloned EJBs</b> .....	18 of 42
<b>Cloning options for the Host Publisher</b>	
<b>JVM</b> .....	19 of 42

<b>Running chained Integration Objects in cloned JVMs</b> .....	20 of 42
<b>Using Host Publisher in a remote open systems environment (OSE)</b> .....	21 of 42
<b>IBM Edge Server for Balancing HTTP Requests</b> .....	22 of 42
Host Publisher Enterprise	
JavaBeans .....	22 of 42
<b>Overview</b> .....	23 of 42
<b>Modifying the Host Publisher EJB</b> .....	25 of 42
<b>Creating EJB support files for Integration Objects</b> .....	26 of 42
<b>Creating a Host Publisher application using EJB Access Beans</b> .....	27 of 42
<b>Using EJB Access Beans to transfer, deploy, and process a Host Publisher application</b> .....	27 of 42
Host Publisher Studio	
Enhancements .....	28 of 42
<b>New Features for Defining a screen</b> .....	28 of 42
<b>Tips for Defining a Screen</b> .....	29 of 42
Express logon .....	31 of 42
<b>Configuring express logon in Host Publisher Studio</b> .....	32 of 42
<b>Configuring express logon in Host Publisher Server</b> .....	32 of 42
<b>Installing Host Publisher 3.5 using express logon on an existing installation of WebSphere Application Server 3.5</b> .....	33 of 42
Appendix A. Performance and tuning .....	33 of 42
<b>System requirements</b> .....	33 of 42
<b>Host Publisher Studio</b> .....	33 of 42
<b>Host Publisher Server</b> .....	34 of 42
<b>Hardware recommendations</b> .....	37 of 42
<b>Server capacity</b> .....	38 of 42

Appendix B: Manually Configuring WebSphere Application Server and your Web server for Host Publisher Server .....	39 of 42
<b>Web Server Configuration</b> .....	39 of 42
Appendix C: Save classpath information from admin.config when installing WebSphere FixPacks .....	40 of 42
Appendix D: Preliminary Check list and Software Installation Checklists for standard installations .....	40 of 42

## Introduction

The target audience for this paper is experienced Host Publisher administrator's and technical personnel that will be implementing applications to run with Host Publisher version 3.5. This paper will focus on important information that you will need to know before upgrading from a previous version of Host Publisher server to Host Publisher Server version 3.5.

It will address software levels, including WebSphere Application Server software and PTFs needed for a successful installation. The information presented here will supplement the product documentation in the Installation guide, and the Administrator's and User's Guide.

This paper will also cover discussion of the major new features in the Host Publisher 3.5. When discussing these new features it is assumed that you are already familiar with the terminology and have prior knowledge of Host Publisher. If you are not already familiar with Host Publisher it is suggested you read the Host Publisher 3.5 Administrator's and User's Guide first.

## New Features Overview

IBM WebSphere Host Publisher is a key component of IBM's Host Integration software portfolio. With this new release several key new features have been added. Host Publisher Version 3 Release 5 includes the following new function and features which will be discussed in more detail later in this paper.

- **Distributed administration and support for WebSphere's load balancing**

Host Publisher has been enhanced to support WebSphere Application Server's load balancing capabilities (cloning of JVMs on one machine and across multiple machines). You can now use one browser to administer multiple cloned Host Publisher instances.

- **Enterprise JavaBeans (EJB) support**

Host Publisher delivery of EJB support provides those who are building EJB-based applications a locatable, scalable, distributed, and manageable server-side Java component with Host Publisher Integration Object function.

- **Express logon feature**

Express logon allows you to log on to a 3270 application using a standard X509 V3 client certificate received through WebSphere Application Server from a browser. Use of the client certificate results in reduced password and user ID administration costs.

- **JavaServer Pages (JSP) 1.0 support**

Host Publisher Studio now supports only JSP 1.0 pages. A migration tool is provided to help you convert .91 JSP tags (created prior to Host Publisher Version 3.5) to JSP 1.0 tags.

- **Host Access advanced screen recognition**

In addition to the ability to recognize screens by text area, cursor position, and number of fields, the following recognition capabilities are now provided.

- Comparison of text case (text area recognition)
- Comparison of text color to a specified color
- Comparison of a region to a value
- Comparison of multiple region values
- Multiple screen recognition criteria can be combined to form a logical expression for a single screen
- Logical NOT

- **Keyboard settings**

Using Host Access, you can configure your host connection to use the keyboard settings you prefer. You can also add additional host functions to our predefined list. (Keyboard settings can be configured only in Host Access.)

- **Recording a macro for a chained Integration Object**

Using Host Access to record a macro for an Integration Object that uses chaining is now easier. If you are recording an Integration Object that is middle or last in chain, you no longer have to manually navigate your terminal screen to your starting point before you begin recording your data macro. Instead, Host Access has a new function called Play Another Macro. Play Another Macro allows you to select another macro, for example, the data macro of the Integration Object that is first in the chain, and play that macro. Your terminal screen will be navigated through your selected macro.

## [Host Publisher Prerequisites](#)

### [Supported versions of WebSphere Application Server](#)

Host Publisher for AIX, iSeries, Solaris, Windows NT, and Windows 2000 requires WebSphere Application Server ("WebSphere").

For development purposes only, a WebSphere Advanced Edition Version 3.5 for Windows NT and Windows 2000 CD is included.

**Note that WebSphere must not be installed using the Quick Installation option because WebSphere does not support use of InstantDB in a production environment.**

Host Publisher 3.5 requires WebSphere Advanced Edition Version 3.5 FixPack3 (3.5.3) applied. Obtain FixPack3 by downloading it from [WebSphere support](#). You must upgrade to

## Host Publisher Version 3.5 White Paper

WebSphere 3.5.3 before Host Publisher can be installed. In addition, before you install Host Publisher, we recommend that you apply the following maintenance to WebSphere 3.5.3:

- PQ44919 - all platforms
- PQ45064 - all platforms
- PQ45976 - Windows NT and Windows 2000
- PQ46615 - all platforms except iSeries
- PQ46762 - all platforms
- PQ46998 - all platforms

### Installing the Web server

For the workstation platforms, WebSphere 3.5 includes a Web server: IBM HTTP Server Version 1.3.12. This Web server is optionally installed for you during the installation of WebSphere 3.5. FixPack 3 for WebSphere 3.5 contains fixes for IBM HTTP Server.

If you have a Web server installed on your system that is not currently supported by WebSphere 3.5.3, and you wish to use it instead of IBM HTTP Server, you should upgrade it to a supported level prior to installing WebSphere. Refer to the WebSphere documentation, or visit the [WebSphere home page](#) for information about supported versions of Web servers.

## Host Publisher Application Migration from Host Publisher 2.x to 3.5

### Migrating JavaServer Pages (JSPs)

Because Host Publisher Version 3.5 no longer supports 0.91 JSPs, migration of pages created by a previous version of Host Publisher is required; however, you can choose when and how you migrate:

- When Host Publisher Server is installed.
- Using the command line utility on the Host Publisher Server.
- When an application using 0.91 JSPs is opened in the Host Publisher Studio.
- When a 0.91 JSP is imported into the Host Publisher Studio.
- Using the command line utility on the Host Publisher Studio.

Host Publisher provides a migration utility called JSPMigrator. For detailed information about the migration of JSPs, see the *IBM WebSphere Host Publisher Administrator's and User's Guide*.

## Host Publisher Version 3.5 White Paper

So it is up to the person installing Host Publisher 3.5 to decide when and how the JSPs will be migrated. When the Host Publisher Server is installing a choice can be made to migrate all JSPs as part of the installation process, or you can elect to not migrate at installation time and migrate as the application is opened in Host Publisher Studio or one could use the command line migration tool. Be aware, however that applications created with a previous version of Host Publisher will not run until migration is complete.

In addition, if you plan to use WebSphere for cloning application servers (JVMs), and you have existing applications using chained Integration Objects, you need to modify your JSPs to cause HTTP session affinity to be established. For more information, see the *IBM WebSphere Host Publisher Programmer's Guide and Reference*.

When migrating from previous versions of Host Publisher to Version 3.5, you must install Version 3.5 in the same directory path to ensure access to your existing applications. Migration from previous versions of Host Publisher to Version 3.5 is not backwards compatible; that is, Integration Objects created with Host Publisher Studio Version 3.5 will not run on a previous version of the Host Publisher Server; however, Host Publisher Integration Objects created by a previous version of Host Publisher Studio will continue to run on any level of Server greater than or equal to the Studio level used to build the Integration Objects.

### **[WebSphere Studio 3.5 and Host Publisher 2.1 Integration Objects](#)**

If you are using WebSphere Studio 3.5 with Integration Objects created using Host Publisher 2.1, they will not open in the WebSphere Studio 3.5 because they are built using an older JDK. To fix this problem, use the instructions to download and run the Bean Migration Utility on a machine that has Host Publisher Studio installed.

The Bean Migration utility is available in a Host Publisher Technote. To find the Technote:

1. From the Host Publisher Support page (<http://www.ibm.com/support/>), select Software under Search by category.
2. Select Networking and Communications for the Product Family, then select Host Publisher for the Product Type.
3. Search for the Bean Migration utility.

### **[Migrating from Host Publisher 2.2 and WebSphere 3.0.2.1 to Host Publisher 3.5 and WebSphere 3.5.3](#)**

#### **[What you must know before installing Host Publisher version 3.5](#)**

Host Publisher 3.5 requires WebSphere Application Server Advanced Edition 3.5 with PTF 3. Prior to installing Host Publisher 3.5 Server software you must ensure that a supported web



## Host Publisher Version 3.5 White Paper

server is installed and operational, and WebSphere Advanced Edition 3.5 with PTF 3 is installed and operational.

Before you begin installing Host Publisher, you must know the following:

- Which webserver(s) you want Host Publisher install to configure, or if you are using Remote OSE, you would choose not to configure any web server.
- The directory path into which you will install Host Publisher  
**Note: If you are upgrading from an earlier version of Host Publisher Server, you should uninstall the previous version prior to installing Version 3.5. You must install Host Publisher Server Version 3.5 in the same directory path to ensure access to your existing applications.**
- The number of Host Publisher licenses (connections to host or database resources) for which you are authorized
- Whether you have existing applications created with a previous version of Host Publisher
- If you have existing applications, and you want to migrate the JSP pages during installation, the directory path to the JSP pages. Refer to “JavaServer™ Page (JSP) migration” for more information.
- The *alias* you want to use (for Host Publisher Server only) The alias is a string that appears in every URL request. It allows the organization of Host Publisher document parts under one directory.
- The desktop folder name (for Windows NT and Windows 2000)

### Hardware and Software Requirements

Host Publisher Studio runs on Windows 95, Windows 98, Windows NT Version 4, and Windows 2000. Host Publisher Server runs on several platforms.

The platforms, and the requirements for each, are listed below.

#### **Hardware requirements**

##### **Host Publisher Server**

**Windows NT** 40 MB of disk space; 256 MB RAM (512 MB recommended).

**Windows 2000** 40 MB of disk space; 256 MB RAM (512 MB recommended).

**AIX** 23 MB of disk space; 512 MB RAM.

**Solaris** 21 MB of disk space; 256 MB RAM (512 MB recommended).

We recommend that you allow 512 MB of swap space (256 MB minimum) and room for 512 file descriptors (256 minimum).

##### **Host Publisher Studio**

**Windows 95, Windows 98, Windows NT, and Windows 2000:**

Pentium ® 366 processor or higher

Minimum of 256 MB RAM

102 MB of free disk space

A display with a color depth of 256 colors or higher, and a minimum screen resolution of 600 x 800 pixels.

**Software requirements**

**Host Publisher Server**

**Windows NT**

Windows NT Workstation or Server Version 4.0, Service Pack 6A

WebSphere Application Server Advanced Edition 3.5, PTF 3

One of the following supported Web Servers:

- Apache Server 1.3.12
- IBM HTTP Server (included with WebSphere Application Server) V1.3.12, V1.3.12.1, and V1.3.12.2
- iPlanet Enterprise Server 4.0 SP6 and 4.1 SP5
- Lotus ® Domino™ Application Server Release 5.02b and 5.05
- Microsoft ® Internet Information Server V4.0

**Note:** An FTPD product must be installed and enabled to transfer applications from the Host Publisher Studio to the Server. On Windows NT, with Host Publisher Server and Host Publisher Studio installed on the same machine, you can use the *localhost* option. See the *IBM WebSphere Host Publisher Administrator's and User's Guide* for more information.

**Windows 2000 Professional**

Windows 2000 Server or Advanced Server

WebSphere Application Server Advanced Edition 3.5, PTF 3

One of the following supported Web Servers:

- Apache Server 1.3.12
- IBM HTTP Server (included with WebSphere Application Server) V1.3.12, V1.3.12.1, and V1.3.12.2
- iPlanet Enterprise Server 4.1 SP5
- Lotus Domino Application Server Release 5.05
- Microsoft Internet Information Server V5.0

**Note:** An FTPD product must be installed and enabled to transfer applications from the Host Publisher Studio to the Server.

On Windows 2000, with Host Publisher Server and Host Publisher Studio installed on the same machine, you can use the *localhost* option. See the *IBM WebSphere Host Publisher Administrator's and User's Guide* for more information.

**AIX**

## Host Publisher Version 3.5 White Paper

AIX V4.3.3 with Service Pack 02

WebSphere Application Server Advanced Edition 3.5, PTF 3

One of the following supported Web Servers:

- Apache Server 1.3.12
- IBM HTTP Server (included with WebSphere Application Server) V1.3.12, V1.3.12.1, and V1.3.12.2
- iPlanet Enterprise Server 4.0 SP5 and 4.1 SP5
- Lotus Domino Application Server Release 5.05

**Note:** An FTPD product must be installed and enabled to transfer applications from the Host Publisher Studio to the Server.

### **Solaris**

Sun Solaris V2.6 or V2.7 SPARC and the Native Threads Package

WebSphere Application Server Advanced Edition 3.5, PTF 3

One of the following supported Web Servers:

- Apache Server 1.3.12
- IBM HTTP Server (included with WebSphere Application Server) V1.3.12, and V1.3.12.1
- iPlanet Enterprise Server 4.0 SP5 and 4.1 SP5
- Lotus Domino Application Server Release 5.05

**Note:** An FTPD product must be installed and enabled to transfer applications from the Host Publisher Studio to the Server.

### **Host Publisher Studio**

**Windows 95, Windows 98**

None

**Windows NT**

Windows NT Workstation or Server Version 4.0, Service Pack 4

**Windows 2000**

Windows 2000 Professional, Server, or Advanced Server

### **Limitations**

Host Publisher Version 3.5 uses IBM WebSphere Application Server V3.5. Refer to documentation on this product to understand any limitations it might have.

### **Upgrading from a Previous version of Host Publisher**

When you have an existing Host Publisher Server installation and want to upgrade it to Host Publisher 3.5, the following steps and information below should be used as a guide to help ensure a successful installation of Host Publisher Server. This procedure assumes that you have met all the above mentioned hardware and software prerequisites. A checklist is provided at the end of this paper that will assist you in completing all the necessary tasks.

## Host Publisher Version 3.5 White Paper

- Uninstall the previous version of Host Publisher. Ensure that WebSphere is running before you start the uninstall process so that the Host Publisher/WebSphere configuration gets removed.
- Upgrade or install a supported Web Server. The list of supported Web Servers is listed above for the workstation platforms. Note: WebSphere Advanced Edition 3.5 includes IBM HTTP Server 1.3.12 that can be installed as part of the WAS AE 3.5 install.
  - ❖ **Note: Ensure that whatever Web server is installed is one that is on the list of supported Web servers and any PTFs that are required are installed. If you are upgrading the existing Web server, ensure that the Web server documentation for migration is followed and after the upgrade that the Web server is functional before proceeding with the next steps.**
- Upgrade to a database that is supported by WebSphere Application Server Advanced Edition 3.5. See <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>
  - ❖ **Note: Use of InstantDB is not supported in a production environment with WebSphere 3.5 Advanced Edition. WebSphere 3.5 Advanced Edition must not be installed using Quick Installation option because WebSphere does not support use of InstantDB in a production environment. WebSphere 3.5 Advanced Edition includes a supported level of DB2 Enterprise Edition V6.1. Based on Host Publisher testing, the recommended levels of DB2 Enterprise Editions are 6.1 FP 4, 5, or 6, or 7.1 FP 3. Follow product documentation for upgrade.**
- Upgrade previous version of WebSphere to WebSphere AE 3.5. Refer to "Migrating from previous product versions" in the WebSphere Infocenter at <http://www.ibm.com/software/webservers/appserv/doc/v35/ae/infocenter/index.html>.
  - ❖ Note: After installing WebSphere Advanced Edition 3.5 verify that it is operational by opening the WebSphere Administrator's Console. Verify it loads successfully.
    - ◆ Start the Default Server from the console. Verify its state changes to "Running".
    - ◆ Open a browser and type in the URL: [http://your\\_host\\_name/servlet/snoop](http://your_host_name/servlet/snoop)Verify that the Snoop Servlet runs before proceeding.
- Install WebSphere Advanced Edition 3.5 PTF 3
  - ◆ Stop both the Web Server service and the WebSphere Advanced Edition AE 3.5 service before installing the WAS 3.5 PTF 3.
  - ◆ Install the PTF 3 according to the readme instructions.
  - ◆ **Important: Reboot the server and verify that both the Web server and the WS AdminServer services are started before proceeding to the next step.**
- Install Host Publisher Server 3.5
  - Reboot the server and continue on to the next section below to test the installation.

### Testing installation of the Host Publisher Server

After you complete installation of Host Publisher Version 3.5, start the Web server, then access a simple HTML page. In this way, you can ensure that the Host Publisher alias is working.

1. Copy the following HTML code to a file named test.html.

```
<!doctype html public "html2.0">
<html>
```

## Host Publisher Version 3.5 White Paper

```
<body>
<h1 align="center">My home page</h1>
</body>
</html>
```

2. Place test.html in the *install\_dir*\Server\production\documents directory path, where *install\_dir* is the directory in which Host Publisher is installed.

On Windows NT and Windows 2000, *install\_dir* defaults to c:\HostPub\.

On AIX and Solaris, *install\_dir* is usually /var/HostPublisher/.

3. Load <http://myhost/alias/test.html> in your browser, where *alias* is the alias you defined when installing Host Publisher. The default alias is /HostPublisher/.

If the test file is not found, check the Web Server's configuration files to ensure the Host Publisher aliases are defined. Stop and restart the Web Server if you make changes. Refer to the information about Web server configuration in the troubleshooting section of the *IBM WebSphere Host Publisher Administrator's and User's Guide for more information about Host Publisher aliases*.

4. Start the Host Publisher Application Server using the WebSphere Administrator's Console application. Host Publisher is configured in its own JVM as its own Application Server.

### Note:

- **Windows NT and Windows 2000:** The Console is started with the adminclient.bat script.
- **AIX and Solaris:** The Console is started with the adminclient.sh script.
- The adminclient.bat file is located in the /WebSphere/AppServer/bin directory.

In the Console, select the HostPubServer Application Server you find beneath your Node, and press the Start icon. A dialog box will indicate when the server has successfully started.

Next, access a simple JSP page to make sure all the basic WebSphere Application Server settings are correct.

A. Copy the following HTML code to a file named test.jsp.

```
<html>
<head>
<title>JSP Test</title>
</head>
<body>
<h2>Hello world</h2>
<h4>I'm at <%=request.getRequestURI() %></h4>
</body>
</html>
```

Host Publisher Version 3.5  
White Paper

B. Place test.jsp in the *install\_dir*\Server\production\documents directory path, where *install\_dir* is the directory in which Host Publisher is installed.

C. Load <http://myhost/alias/test.jsp> in your browser, where *alias* is the alias you defined when installing Host Publisher. The default alias is /HostPublisher/.

If <http://myhost/alias/test.jsp> fails to load, ensure that WebSphere Application Server is started. Ensure that the Application Server named HostPubServer is started in the WebSphere Administrator's Console.

5. Next, check the configuration of Host Publisher to WebSphere Application Server.

A. Start the WebSphere Administrator's Console.

B. Ensure that the **HostPubServer**, **HPServletEngine**, **hostpublisher**, and **\_IBM\_HP\_WebAdmin\_** entries appear in the topology tree.

If these entries are not present, manually run the configuration script located in *install\_dir*, where *install\_dir* is the directory in which Host Publisher is installed.

To run the configuration script, type the following on the command line:

**Windows NT and Windows 2000**

*HostPubWAS.bat HostPubWASConfig.xml*

**AIX and Solaris**

*./HostPubWAS.sh HostPubWASConfig.xml*

C. If you want to use EJB support, ensure that the **HPubEJB** entry appears in the topology tree in the **Default Container** of **HostPubServer**.

If this entry is not present, manually run the configuration script located in *install\_dir*\Common\EJB\HPubEJB10, where *install\_dir* is the directory in which Host Publisher is installed. To run the configuration script, type the following on the command line:

**Windows NT and Windows 2000**

*installEJB.bat*

**AIX**

*./installEJBAIX.sh*

**Solaris**

*./installEJBSUN.sh*

Load <http://myhost/alias/showCfg> in your browser (showCfg is case sensitive), where *alias* is the alias you defined when installing Host Publisher, to see more configuration information. The default alias is /HostPublisher/.

D. Next start the Host Publisher Administration console that will load in a browser. Start the Host Publisher Administration and verify it loads in the browser successfully. You should be able to access all the administration panels and verify the the Host Publisher server status is **Started**.

## Host Publisher Distributed Administration

Distributed administration allows you to perform administration tasks on instances of Host Publisher Server running either in JVM clones defined using WebSphere, or in JVMs with an EJB container but no Web container.

### Accessing Host Publisher Server Administration

The Host Publisher Server provides the runtime environment for supporting Web applications created with the Host Publisher Studio. The runtime environment includes Web-based server administration for controlling the runtime and the applications it serves. To access Server Administration, load this URL in your browser: **http://server/hp\_alias/HAdmin/main.jsp** (where *server* is your server name and *hp\_alias* is the alias chosen during installation of Host Publisher Server).

In Host Publisher Version 2.X, the administration task required an instance of HAdminServlet in every WebSphere JVM in which Host Publisher Server ran. The servlet managed the Host Publisher Server instance in that JVM by making local method calls to the server objects and classes. This is no longer possible in WebSphere 3.5, the current version of WebSphere, because:

- When cloned JVMs are defined within or across machines, an attempt to access the URL for the Host Publisher administration servlet cannot be directed to a specific JVM, because every JVM is identical. The request is not necessarily routed to the desired JVM.
- A Host Publisher user can run Integration Objects in a JVM that contains only an EJB container, with no Web container. However, HAdminServlet cannot run in such a WebSphere JVM.

To overcome these limitations, the Host Publisher Server administration function in Host Publisher Version 3.5 consists of two components:

- A Java Remote Method Invocation (RMI)-based Host Publisher *administration server* that is created in every Host Publisher Server instance that is initialized in a WebSphere JVM.
- An *administration client*, HAdminServlet, that can run in any WebSphere JVM and can manage a Host Publisher Server instance in another WebSphere JVM by making RMI calls to its administration server. HAdminServlet is configured within the Host Publisher Application Server JVM (HostPubServer) which is defined, using WebSphere configuration, when you install Host Publisher Server.

## Host Publisher Version 3.5 White Paper

Most functional and user-interface aspects of Web-based administration have not changed significantly since Host Publisher Version 2. Before you can perform any administration tasks, however, you must select a Host Publisher Server instance to administer on a given server machine. The default is the Server instance running on the same JVM as HPAdminServlet. After you select the Server instance, all administration functions are assumed to apply to that Server instance until the selection is changed.

The following categories of administration tasks are allowed for the Host Publisher Server instance you are managing:

- Selection of a server, and an instance of Host Publisher Server in that server, to administer
- Monitoring server status: starting and stopping Host Publisher Server, and supplying passwords during startup
- License management: changing the allowed number of licenses on a server running one or more instances of Host Publisher Server (JVMs)
- Connection and connection pool administration: displaying pool definitions, pool status, and the status of active connections for Host Publisher Server
- Management of user lists
- Application administration: deploying applications, deleting applications, and deleting files that are no longer needed
- Problem determination: viewing trace and log files, and setting various options for tracing and logging
- XML Legacy Gateway administration: configuring the XML Legacy Gateway to access hosts and host applications, and creating a portal for accessing hosts and host applications.

### [Securing access to Host Publisher Server Administration using WebSphere Application Server](#)

You can protect a WebSphere Application Server administrative domain that consists of a cluster of servers using the same WebSphere Application Server administrative database. By doing this, certain Host Publisher administrative operations are also protected.

To enable WebSphere Application Server 3.5 user ID and password security:

- On the WebSphere Administrative Console menu bar, click Console > Tasks > Configure Global Security Settings.
- On the General tab of the Set Global Security Wizard, select Enable Security.



## Host Publisher Version 3.5 White Paper

Refer to the WebSphere Application Server documentation for information on other security tasks.

After you complete these steps and load `http://server/hp_alias/HPAdmin/main.jsp` in your browser, a logon prompt appears. To run the servlet, type the user ID and password that you created using the previous procedure.

When you use Host Publisher Server Administration to administer a host or JVM (running the Host Publisher server) on a protected WebSphere Application Server, you must provide a user ID and password. This user ID and password must match the user ID and password used for authentication on that WebSphere Application Server.

When WebSphere Application Server security is enabled, the following Host Publisher Server Administration operations require the WebSphere user ID and password.

- Start server
- Restart server
- Shut down server
- Deploy application
- Remove application
- Delete files
- Change the number of licenses
- Delete a connection
- Update passwords on a user list

### [JVM cloning and load balancing in WebSphere](#)

WebSphere Application Server provides support for running multiple JVMs (referred to as application servers in the WebSphere documentation) on a single machine. Each JVM consists of two distinct execution environments: a *Web container* for executing Web applications such as servlets and JSPs, and an *EJB container* for executing Enterprise JavaBeans (EJBs). These JVMs can be *clones* of each other—that is, they consist of the same EJBs and Web applications. For Web applications, all JVM clones can process the same URL request. Alternatively, the JVMs can contain unrelated sets of applications and, for Web applications, they can process different sets of URLs.

JVMs that are clones of each other can be limited to one machine (vertical cloning), or be spread across multiple machines (horizontal cloning). Both types of cloning provide two benefits:

- Improved throughput, because requests can be distributed across multiple JVMs

- Better fault tolerance, because when one JVM fails a user still can request that an application be processed by any of the remaining clones that can handle the request.

Vertical cloning is also useful when configured on a machine which is powerful enough that a single JVM cannot effectively utilize its CPU power (such as on a multiprocessor machine).

WebSphere provides load balancing mechanisms for forwarding a client's requests across multiple cloned JVMs to distribute the load on individual JVMs evenly. There are two different load balancing mechanisms: one for cloned Web applications, and one for cloned EJBs.

### **Load balancing of cloned Web applications**

HTTP requests directed to servlets or JavaServer Pages (JSPs) running in WebSphere are received by a Web server and forwarded to the WebSphere *plugin*. A plugin is a piece of code that "hooks into" a Web server using a standard Web server-specific programming interface. The WebSphere plugin forwards HTTP requests to a given application server (JVM) using a proprietary protocol called OSE.

OSE uses a variety of inter-process communication mechanisms such as pipes and TCP/IP. OSE is integrated with the load balancing feature of WebSphere. If multiple clones are defined to the plugin to respond to a particular URL, the OSE transport automatically distributes the load across the clones. When OSE runs over TCP/IP connections where the plugin and the JVM are in separate machines, it is referred to as remote OSE. Remote OSE is useful in configurations where Web server machines must be separated from machines running WebSphere Application Servers. For more information about OSE, refer to "[Using Host Publisher in a remote open systems environment \(OSE\)](#)".

### **Load balancing of cloned EJBs**

EJB requests directed from a client to an EJB are also load balanced. This type of load balancing is referred to as *Workload Management (WLM)*. Load balancing of EJB requests is performed by the "smart" stub code that runs on the EJB client to perform the remote invocation of the EJB. This topic is discussed at length in the IBM Redbook *WebSphere Scalability: WLM and Clustering Using WebSphere Application Server Advanced Edition*, SG24-6153.

### **Cloning options for the Host Publisher JVM**

Because the Host Publisher Server and Host Publisher Integration Objects can execute in any WebSphere JVM, either in a Web container or an EJB container, they exploit all the benefits of WebSphere load balancing and WLM. The Host Publisher installation process configures a separate WebSphere Application Server in the machine where Host Publisher is installed. The WebSphere application server is called *HostPubServer* and includes two Web applications: *hostpublisher* runs applications generated by Host Publisher Studio, and *\_IBM\_HP\_WebAdmin\_* runs server-specific applications. Because WebSphere allows you to run multiple JVMs on a machine, you can create several variations of this basic configuration. You can:

- Vertically clone the basic HostPubServer JVM within a machine to utilize multiple CPUs.
- In addition to vertically cloning the HostPubServer JVM, add horizontal cloning across machines for scalability and fault tolerance.
- Partition the JVMs in a noncloned setup so that they process different URLs representing different Host Publisher applications. For example, if the two JVMs are called JVM1 and JVM2, the hostpublisher application in JVM1 can be configured to respond to the URL `.../HostPublisher/application1`, and the hostpublisher application in JVM2 can be configured to respond to the URL `.../HostPublisher/application2`. This provides a static form of load balancing based on the request pattern. It also protects each JVM from application-related problems in the other JVM.

Special considerations for running chained Integration Objects in a cloned configuration are covered in a section below titled [Running chained Integration Objects in cloned JVMs](#).

### [Horizontal Cloning Considerations](#)

When using horizontal cloning one or more of the clones will not have a web server installed. On the non web server Host Publisher machine, make sure that during the installation of the Host Publisher server software that you do not check a web server to configure during the Host Publisher server install.

In addition, when using horizontal cloning, on the machines that clones are going to be created on, don't allow Host Publisher install to configure the WebSphere application server. This can be accomplished by making sure that the WebSphere AdminServer service is not started at install time, or the commands HostPubWASStop and HostPubWasDel can be used.

### [Running chained Integration Objects in cloned JVMs](#)

#### [Using HTTP session affinity](#)

Host Publisher applications consisting of chained Integration Objects, which are being driven by servlets and JSPs, depend on a WebSphere feature called HTTP session affinity. The use of this feature is described in the section "Running chained Integration Objects in cloned JVMs" in Chapter 5 of the *Administrator's and User's Guide*.

***Note: To use the HTTP session affinity feature, you must enable HTTP Session Persistence in WebSphere. Refer to WebSphere documentation for details on how to enable HTTP Session Persistence.***

When a URL for a servlet or JSP is received by the Web server and forwarded to the WebSphere plugin, the plugin normally forwards the request to any of the cloned JVMs that have been configured to process that URL. However, when Websphere's HTTP session affinity feature is enabled (it is enabled by default), and the browser has included a *session ID* in the HTTP request,

the normal load-balancing behavior of the plugin is altered. A session ID in an HTTP request is either present in a cookie or included in the URL in an encoded form if URL rewriting is used.

To enforce HTTP session affinity, the plugin maps the session ID to one of the JVM clones in such a way that a given session ID is always mapped to the same clone. Whenever a browser includes a given session ID on its HTTP request, the request is routed to the same clone. Thus, Websphere's session affinity feature creates an affinity between a browser with a given session ID and a given JVM among the clones that can handle that request.

A set of chained Integration Objects share a connection to a host application. This connection is internally represented as a Java object which is valid only in the JVM in which it is created. Because this object cannot be serialized, it cannot be written to a file or database by Host Publisher in one JVM and then recreated and used in another JVM. Therefore, execution of all the Integration Objects in a chained application must occur in the same JVM. You can ensure that this will happen by enabling Websphere's HTTP session affinity feature.

Chained Integration Objects executing in a Web container use the *HTTP session object* - a standard, servlet API-defined object used to track a given browser across HTTP requests - to correlate a browser with the host connection being used by the chained Integration Objects. If an HTTP session object is not already present, the first Integration Object in the chain creates it. This ensures that WebSphere creates a session ID and returns it to the browser along with the HTTP response. The browser returns this session ID on subsequent requests to execute Integration Objects, and the session ID is used by WebSphere to direct those requests to the JVM that contains the host connection object for that Integration Object chain.

When a set of chained Integration Objects are executed in an EJB container using an instance of the Host Publisher stateful session EJB, and the EJB Access Beans themselves are executing in a Web container on behalf of a browser, then HTTP session affinity does not have to be configured in that Web container even if the EJB Access Beans execute in cloned JVMs. This is because whenever a request to execute an Integration Object is sent by an EJB Access Bean to the Host Publisher stateful EJB, the request includes the handle of that EJB instance. This unique handle guarantees that the EJB protocols will direct the request to execute the EJB to the JVM running that EJB instance. Therefore, all chained Integration Objects executed by the Host Publisher EJB instance on behalf of a browser execute in the same JVM.

EJB Access Beans executing in Web containers use the HTTP session object associated with a browser to store and track the handle of the EJB instance being used to run Integration Objects for that browser. Therefore, in this configuration, *HTTP session persistence* must be configured if HTTP session affinity is not enabled. HTTP session persistence is a WebSphere feature that allows HTTP session objects to be serialized to a database by one JVM and recreated from that database in another JVM. This feature allows all EJB Access Beans running in cloned JVMs to access the EJB handle associated with a browser running chained Integration Objects.

The following section describes how to use remote OSE to set up cloned JVMs running Host Publisher, (that is, to separate the Web server from the machines running WebSphere and Host Publisher.)

## Using Host Publisher in a remote open systems environment (OSE)

WebSphere Application Server uses a proprietary protocol called OSE to communicate between a plugin and the servlet engine. When you run both the Web server and WebSphere on the same machine, OSE is usually run over local pipes. If the Web server and WebSphere are running on different machines, however, you can use remote OSE over TCP/IP.

**Note: When using remote OSE with Host Publisher, at Host Publisher install time you do not need to choose a web server to configure. Leave all the check boxes blank if the web server is remote.**

Remote OSE allows the Web server to run on a separate machine and send requests to one or more application servers running on remote machines. OSE supports clustering and workload management of application servers. This means that the Web server can send requests that require intensive processing to multiple application servers, freeing up the Web server to process more requests.

Refer to WebSphere documentation for detailed information about the administrative tasks required to configure a plugin for Remote OSE and remote application server machines.

In a remote OSE configuration, the Web server and application server run on different machines. When the WebSphere plugin receives a request from the Web server, it determines if the request should be forwarded to the application server, or if it should be passed to the Web server on the local machine. Requests for pages that contain dynamic content, such as JavaServer pages (JSPs), are forwarded to the application server. Requests for pages that contain only static content are passed to the Web server. Since Host Publisher applications can contain both static and dynamic pages, some manual configuration is required on the Web server in a remote OSE configuration.

To set up the Web Server and the default Host Publisher alias, do the following:

Create a directory for pages that contain static content, for example, on Windows platforms:.

```
md c:\HPStatic
```

Add Host Publisher aliases; for example, on Windows platforms:

```
Alias /_IBM_HP_WebAdmin/ "C:/HPStatic/"  
Alias /HostPublisher/ "C:/HPStatic/"
```

Copy the Host Publisher application directories from the Host Publisher /Server/production/documents directory on the application server machine to the HPStatic directory on the Web server machine. For Windows platforms copy directories from c:/Hostpub/Server/production/documents to c:/HPStatic.

Host Publisher has two built-in applications (Host Publisher Server Administration, and XML Legacy Gateway) that require both static and dynamic pages. To enable these functions, copy the HPAdmin and xmlLegacyGateway directories from the /production/documents directory to the HPStatic directory. Remember that when new Host Publisher applications containing static pages

are deployed, the directories for these applications must be copied to the HPStatic directory on the Web server machine.

## **IBM Edge Server for Balancing HTTP Requests**

IBM WebSphere Edge Server is a complete product approach to address Web site response time, scalability, and reliability. The Edge Server provides an integrated solution for local and wide-area load balancing, content-based quality of service routing, and Web content filtering and caching for multi-vendor Web server environments.

If you need to load balance HTTP requests incoming to a Host Publisher server or cloned JVMs where Host Publisher applications are running, a recommended solution is to use IBM Edge Server. For more information about the IBM Edge Server refer to the following:  
<http://www-4.ibm.com/software/webservers/edgeserver/>

## **Host Publisher Enterprise JavaBeans**

The addition of Host Publisher EJB support provides a new option for running Integration Objects.

It is meant for customers that are running their back-end servers using EJB technology and want to logically (or physically) partition the Integration Object's function of navigating and retrieving back-end enterprise data from the presentation/consumption of that data. This aligns Host Publisher with the J2EE 3-tier application model.

The web application portion (JSP/Servlet) can run on a web-tier and remotely locate and access the Host Publisher EJB to process any EJB-enabled Integration Object or chain of Integration Object's. The HPub EJB is 'WLM-ized' by default to take part in Websphere's work load management function. The HPub EJB is implemented as a Stateful EJB to run Integration Object chains, and is resource managed by the EJB Container including life cycle management with passivation and activation of EJB instances when needed.

Enterprise JavaBeans (EJBs) is a server-side component architecture that enables rapid development of versatile, reusable, portable applications.

EJB technology delivers benefits that address the concerns of enterprise development teams, such as:

- Reduced time to market for mission-critical applications
- Enhanced scalability and portability
- Reduced development cost of enterprise scale applications

## Host Publisher Version 3.5 White Paper

Host Publisher Server now provides a Host Publisher EJB deployed in a WebSphere Application Server that is used to run any EJB-enabled Integration Object or Integration Object chain in an EJB environment.

Host Publisher Studio's Host Access and Database Access now include the option to generate EJB support for Host Publisher Integration Objects.

### **Note:**

EJB support can be generated for all Integration Objects except for those configured to use the express logon feature.

### **Overview**

Host Publisher provides support for executing Integration Objects in EJB containers to take advantage of the server-side characteristics provided by the Enterprise JavaBeans architecture. This support consists of the following parts:

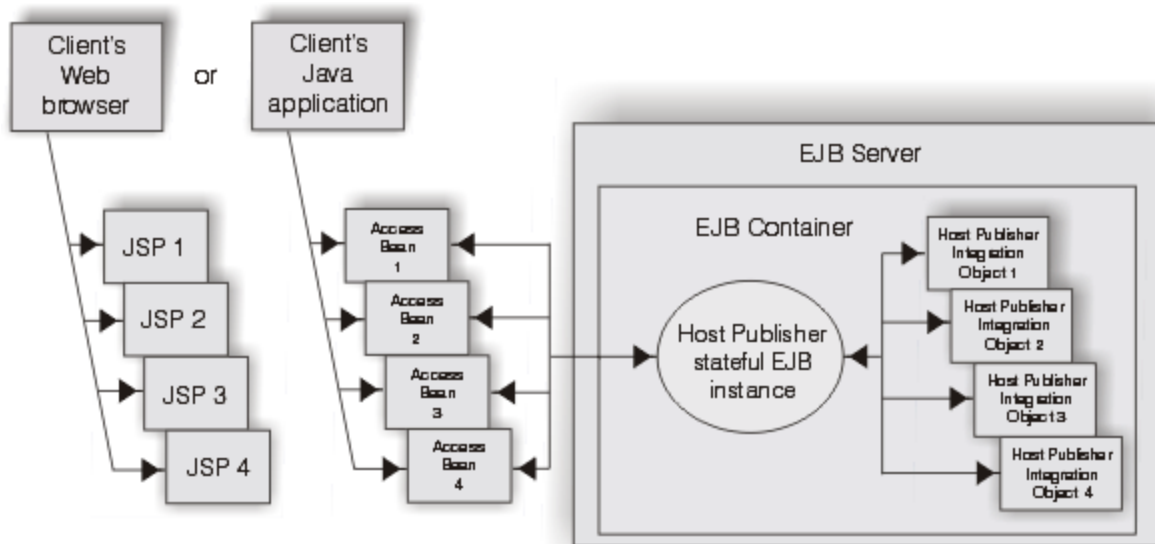
- The Host Publisher EJB, which is a stateful EJB capable of running Integration Objects in an EJB environment. This EJB is installed in the Host Publisher application server Default Container of WebSphere when Host Publisher Server is installed.
- Host Publisher Studio support for the generation of EJB support files for running Integration Objects. This includes the generation of an EJB Access Bean for each Integration Object, that provides the same signature as the "real" Integration Object.
- Host Publisher Studio support for building applications using the generated EJB support files.
- Since the EJB Access Bean has the same signature as the Integration Object, the EJB Access Bean can be used in client-side code exactly as the "real" Integration Object would have been used. Therefore, the client can be:
  - A JSP or a servlet that uses one or more EJB Access Beans in a Web application where the Integration Objects execute in an EJB container.
  - A custom EJB that uses one or more EJB Access Beans to execute the Integration Objects in an EJB container.
  - A Java application that runs outside a WebSphere execution environment.

Execution flow is as follows:

- The EJB access bean communicates with the Host Publisher stateful session EJB using the remote method invocation/Internet InterORB Protocol (RMI/IIOP), passing it the input properties, and invoking its execution method.

Host Publisher Version 3.5  
White Paper

- The session EJB executes the "real" Integration Object and packages its output properties, passing it back to the access bean as the result of the execution method.
- These output properties are used to generate values for the output properties of the access bean, which can be subsequently used in a JSP, servlet, custom EJB, or a Java application.



This above figure illustrates these concepts where the JSPs are executed in a Web container, the client Java application in a non-WebSphere JVM, and the Host Publisher EJB in a WebSphere EJB container.



## Modifying the Host Publisher EJB

During the installation of Host Publisher Server, the Host Publisher EJB is installed in the default container of Host Publisher Application Server, and named HPubEJB or HPubEJBHome. This installed Host Publisher EJB is sufficient to process all EJB-enabled Integration Objects within an EJB environment. However, you might want to modify the Host Publisher EJB and deploy this modified EJB in other environments.

For example, you might want to:

- Prepare a Host Publisher EJB to be deployed in an EJB container other than the one located within Host Publisher Application Server.
- Change some of the default EJB values specified for the installed Host Publisher EJB.

During installation of Host Publisher Server an EJB input .jar file (HPubEJB.jar) and an .xml input file (HPubEJB.xml) are placed in the *install\_dir*\Common\EJB\HPubEJB10 directory. The WebSphere Application Server jetace tool is used to customize and create a deployable EJB .jar file using these files. Refer to WebSphere documentation for details regarding the usage of the WebSphere jetace tool, as well as a description of deploying and creating an EJB within a WebSphere Advanced Edition container.

The following .jar files must be in the classpath when invoking the WebSphere jetace tool to process the Host Publisher EJB input .jar file:

*Hostpub\_install\_dir*\Common\HpRte.jar

*Hostpub\_install\_dir*\Common\HPubCommon.jar

*WebSphere\_install\_dir*\lib\servlet.jar

The following are Host Publisher attributes you might want to modify when preparing and creating your custom Host Publisher EJB:

### **JNDI HomeName**

This is the network name with which the Host Publisher EJB is registered in the JNDI directory and is used by the client (the Access bean) to locate the EJB. In most cases the default name is appropriate; however, if you want to separate the processing of various Host Publisher EJB requests, you can create multiple deployable Host Publisher EJBs, each with a different JNDI name. Each of these EJBs are separately deployed in different WebSphere Application Server containers on one or more workstations. The JNDI HomeName for the installed Host Publisher EJB is *\_com\_ibm\_HostPublisher\_EJB\_HPubEJB*.

### **Session Timeout**

Specify the number of seconds of inactivity of a bean instance before it times out. In most cases, the default value is appropriate; however, you can specify a different value that is specific to your applications and work environment. The session timeout value for the installed Host Publisher EJB is 600.

### **State Management Attribute**

The installed Host Publisher EJB is a stateful session bean. This is necessary to process chained Host Access Integration Objects; however, if your applications do not process chained Integration Objects, you can specify a stateless session bean to take advantage of the increased performance characteristics associated with stateless session beans.

## **Creating EJB support files for Integration Objects**

Host Publisher Host Access and Host Publisher Database Access now include an option to create EJB Integration Object Support. In both Host Access and Database Access, select Options from the toolbar, then select Create EJB 1.0 Integration Object Support. This option must be checked to generate EJB support files for Integration Objects.

The Studio Options menu contains an EJB Integration Object Properties selection. A default suffix is provided; however, you can specify a different suffix for each EJB Integration Object you create. If you open a different Integration Object, the suffix defaults to the value you assigned previously to that Integration Object.

To change the default value, used by new Integration Objects, for these properties, edit the following properties in Studio.ini:

- EJB\_PROPERTIES\_SUFFIX
- EJB\_HELPER\_SUFFIX
- EJB10\_ACCESS\_SUFFIX

When you change a suffix, you change the suffix of only the Integration Object on which you are working.

When you choose to create EJB Integration Object support, EJB support files are generated during creation of the Integration Object. An EJB Access Bean .jar file is created in the IntegrationObjects directory, and the Integration Object .jar file has additional EJB support class files added to it. The default name for the EJB Access Bean .jar file is *IOName*Access0.jar, where *IOName* is the name of the Integration Object.

### **Note:**

An EJB Access Bean properties file (for example, *IOName*Access0.properties) is included in the EJB Access Bean .jar file. This .properties file specifies parameter values used by the EJB

## Host Publisher Version 3.5 White Paper

Access Bean, including host name and port number of the JNDI server and the JNDI name of the Host Publisher EJB to connect to. The default values assume a local JNDI server and a default Host Publisher EJB JNDI name of `_com_ibm_HostPublisher_EJB_HPubEJB`.

To change the default values:

- Edit the `Hostpub_Install_Dir\Studio\IntegrationObjects\EJB\AccessEJB.properties` file prior to generating the Integration Object (this file is used as input to create the `.properties` file in the EJB Access Bean `.jar` file)

or

- Replace the `.properties` file in the EJB Access Bean `.jar` file with a modified copy after the `.jar` file is generated.

### [Creating a Host Publisher application using EJB Access Beans](#)

After EJB Access Bean `.jar` files are created for Integration Objects, you can use them with the Host Publisher Studio in the same way that Integration Object `.jar` files are used to create Host Publisher applications.

You can create JSPs to instantiate, drive and render the EJB Access Bean output properties in the same way that the original Integration Objects are used, including support for EJB Access Beans created for chained Integration Objects.

### [Using EJB Access Beans to transfer, deploy, and process a Host Publisher application](#)

#### **Note:**

The following information applies only when the Host Publisher application is running on the same server where the Host Publisher EJB is deployed and active. To run the EJB on a different workstation, see further information in the Administrator's and User's Guide, section, [Running the EJB Access Bean on a workstation other than where the Host Publisher EJB is active](#).

Transfer and deployment of Host Publisher applications that are created using EJB Access Beans is the same as transfer and deployment of Host Publisher applications using Integration Objects.

To process Host Publisher applications using EJB Access Beans, you must add the original Integration Object `.jar` files used by the application to the classpath of the application server where the Host Publisher EJB is deployed and active.

To do this:

## Host Publisher Version 3.5 White Paper

In the WebSphere Application Server Administrative Console, highlight the application server entry.

In the Command line arguments field (located in the right pane), add a `-classpath` parameter specifying the fully-qualified Integration Object .jar file, or add the fully-qualified Integration Object .jar file to an existing `-classpath` parameter. For example, if Host Publisher Server is installed at `D:\Hostpub` and you have an Integration Object named `Fulist.jar`, for Windows NT, type **`-classpath D:\Hostpub\Server\production\beans\Fulist.jar`**; in the input field, or add `D:\Hostpub\Server\production\beans\Fulist.jar`; to an existing `-classpath` parameter.

Click Apply.

**Note: For WebSphere Application Server 3.5 Advanced Edition, changes to application server command line arguments do not become active until you stop and restart the application server.**

To run the EJB Access Bean on a workstation other than where Host Publisher EJB is deployed and active, modify the EJB Access Bean .properties file to specify the location of the JNDI directory in the network, then copy the modified EJB Access Bean .jar file to the target workstation.

The following WebSphere Application Server 3.5 Advanced Edition and Host Publisher .jar files are required on the target workstation:

- *WebSphereInstallDir*\AppServer\lib\ujc.jar; (for EJB client code)
- *WebSphereInstallDir*\AppServer\lib\ejs.jar; (for EJB client code)
- *HostPublisherInstallDir*\Common\HPubCommon.jar

In addition, copy the deployed EJB .jar file to the target workstation; for example, for Windows NT:

*WebSphere\_install\_dir*\AppServer\deployedEJBs\\_wlm\_DeployedHPubEJB.jar

**Note:**

These .jar files must be present in the classpath to run the EJB Access bean.

## **Host Publisher Studio Enhancements** **New Features for Defining a screen**

Defining a screen provides a way for the macro to identify that it has reached the desired host screen. When the macro plays, it waits until the screen is recognized as defined before playing more keystrokes. (Each step in a macro has at least one screen defined as the possible next screen; Host Publisher waits for the expected screen to appear before playing the actions associated with the next step of the macro.)

When you are prompted to define a screen, before you can enter more information on the host terminal screen, you must either define the screen or respond that you do not want to define the screen. If you choose not to define the screen, the screen is unrecognized. When the completed macro plays an unrecognized screen, instead of waiting for the expected screen, the recorded keystrokes play regardless of which terminal screen appears.

In addition to the ability to recognize screens by text area, cursor position, and number of fields, the following recognition capabilities are provided.

- Comparison of text case

Case sensitive is selected by default.

- Comparison of text color to a specified color

You can change the coordinates of the color region and select different background and foreground colors.

- Comparison of a region to a value

You can change the start and end position coordinates of the defined region on the terminal screen.

- Comparison of two region values

You can change the start and end position coordinates of the first and second regions you defined on the terminal screen.

- Multiple screen recognition criteria can be combined to form a logical expression for a single screen

You can choose whether each criterion is optional or non-optional.

- Logical NOT

You can match a specified criterion or not match a specified criterion.

### **Tips for Defining a Screen**

Some tips for defining a screen in a macro are:

Never define a screen by text that could change over time, such as a connection identifier, date, time stamp, or user ID.

Define a screen using specific criteria to ensure that only one screen matches any set of next screens.

## Host Publisher Version 3.5 White Paper

Define a unique string that can be located anywhere on the screen without specifying its location; this eliminates the possibility of the string being in a different position and your screen not being recognized.

Some screens arrive in segments, depending on the complexity of the screen.

If you specify a screen description that identifies a part of the screen that arrives in one of the first segments, Host Publisher might recognize the screen and start performing the actions before the complete presentation space on the terminal has been updated.

Therefore, make sure you always specify a detailed screen description to ensure that the last screen segment arrives before the actions in the tag are executed.

To find out if your host application screens are segmented, turn on tracing in your host emulator or on the TN3270 server.

Notes:

Screen segmentation is controlled by the host application and not by SNA parameters such as RU sizes or DLC MTU sizes. In SNA, a screen segment corresponds to a chain.

If you cannot determine if a screen is segmented, and you want to reliably recognize the screen in a macro, you can adjust the macro's *pause* parameter. Pause is the time delay, in milliseconds, between when actions of a screen description are performed and when valid next screens of the screen description are registered to the Host On-Demand screen recognition logic. The pause default is 200 milliseconds. This attribute must be modified using a text editor. For example:

```
<screen name="ready.2" entryscreen="true" exitscreen="false"  
  transient="false" pause="3000">
```

If the host application sends various screen segments in response to inbound keystrokes sent during actions, this delay can be used to ensure that all segments of the screen update arrive before the next screen match is attempted. Pause introduces a delay in the macro execution which affects performance and response time. If no application screens seen by a macro are segmented, setting pause to zero might result in improved Host Publisher runtime performance and scalability.

There is a global *pausetime* attribute in the first tag in the macro that sets the default pause time for all screens in that macro. You can override this attribute by modifying the pause parameter.

Define global screens and associated actions for those screens that might appear at any time, such as monthly reminders or messages from colleagues. Global screens allow you to define a generic action to take (such as a clear screen command) if such a screen is encountered, allowing the macro to return to the normal flow.

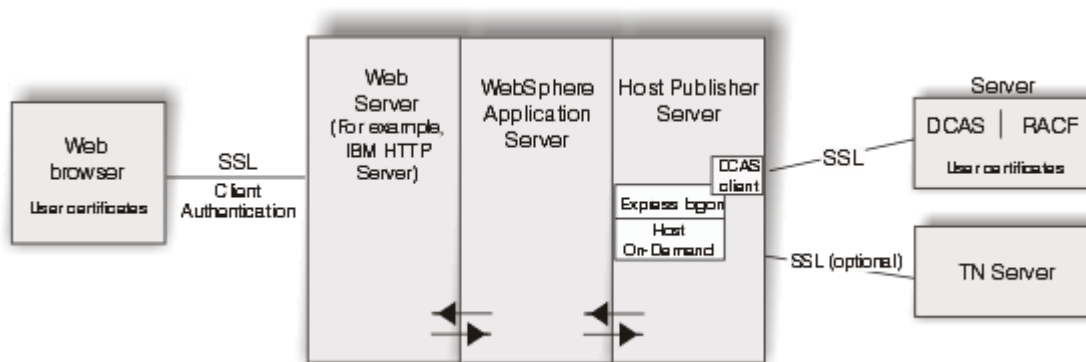
If you use Host Access to record your macro, you cannot navigate back through a screen definition. To change the definition, you must right-click on the screen node in the macro tree and select Modify.

## Express logon

Express logon allows a user with a Web browser certificate to log on to a host system through a Web browser without having to enter the user ID and password. This function is designed to reduce the time spent by an administrator maintaining host user IDs and passwords. It also is designed to reduce the number of user IDs and passwords that users have to remember.

In Host Publisher, express logon allows a macro to log on to a host application without the browser user having to enter the user ID and password. All interaction with the host application is performed by macros executing on the Server. Host Publisher uses express logon to log on to host applications using client certificates obtained from a browser.

### Express logon



The express logon architecture describes the following key components:

- A Digital Certificate Authentication Server (DCAS) component, provided by z/OS, that communicates with an enhanced version of the host-based RACF facility for user ID and password administration. DCAS dynamically supplies a user ID and a one-use-only passticket based on an X.509 certificate and the ID of the host application to which you want to log on.
- A DCAS client that communicates with the DCAS server over a client-authenticated SSL connection, performing a proprietary protocol to perform the function mentioned above.

Host Publisher provides a Java-based DCAS client. To use Host Publisher's express logon capability, Integration Objects must be executed by JSPs or servlets, and the Host Publisher Integration Objects must be enabled for express logon in the Host Access application. When the application containing the express logon-enabled Integration Objects executes on Host Publisher Server:

## Host Publisher Version 3.5 White Paper

- When a Host On-Demand macro is executed to a point where you specified to insert a user ID or password using Host Access, Host Publisher Server uses express logon to provide the macro with a user ID or password.
- Host Publisher Server accesses the Web browser's client certificate using a servlet API and retrieves the application ID from the macro you recorded in the Host Access application. You defined the application ID in Host Access when you chose to insert a user ID in your macro. Host Publisher's DCAS client passes the certificate and application ID to the DCAS server and the DCAS server provides the user ID and password requested by the macro.

**Note:** The browser must use HTTPS, and you must configure the Web server to request client authentication.

The connection between the DCAS client and the DCAS server must be a client-authenticated SSL connection. Therefore, to execute express logon-enabled applications in Host Publisher Server, you must use IBM Key Management to create a keyring database, a password-protected Java class file that stores the X.509 certificate used for the client authentication of this connection. You must name the keyring database HostPubELF.class and save it in *install\_dir/Common*. On non-Windows server platforms, you must create HostPubELF.class in Host Publisher Studio and transfer it to /Common in Host Publisher Server.

The following host applications support express logon: TSO, CICS, IMS, and NetView. Any application using RACF for logon validation is a candidate for express logon.

### [Configuring express logon in Host Publisher Studio](#)

If you enable express logon in the Host Access application, when you choose to insert a user ID or password, you must define an application ID and a user ID or password. The application ID is used when you run your Integration Object on Host Publisher Server. It is not used when recording or playing your macro in Host Publisher Studio. The user ID and password are only for recording the macro in Host Publisher Host Access. When you play the macro in Host Access, you must supply the user ID and password. When you run the macro on Host Publisher Server, the user ID and password are retrieved from the Digital Certificate Access Server (DCAS).

### [Configuring express logon in Host Publisher Server](#)

#### **Web browsers and Web servers**

Your Web server requires client authentication from your Web browser. Refer to your Web browser's documentation for information on how to set up certificates for your Web browser. Refer to your Web server's documentation to set up client authentication over the SSL session.

#### **IBM Key Management**

IBM Key Management is a tool you can use to manage your digital certificates. With IBM Key Management, you can create a new key database or a test digital certificate, add Certificate



## Host Publisher Version 3.5 White Paper

Authority (CA) roots to your database, copy certificates from one database to another, request and receive a digital certificate from a CA, set default keys, and change passwords.

### **DCAS server**

For more information on how to configure the DCAS server for express logon, refer to the *Setting up and Using the IBM Express Logon Feature white paper* at <http://www-4.ibm.com/software/network/library/whitepapers/elf.html>. Refer to Part 2: *Configuring the Express Logon Feature (ELF)*.

### **RACF**

You must register all Web browser client certificates with RACF. This associates the certificates, which are passed by Host Publisher Server to the DCAS server, with the IDs of users attempting to log on. For more information on RACF commands, refer to *OS/390 SecureWay Security Server RACF Security Administrator's Guide and OS/390 SecureWay Security Server RACF Command Language Reference*.

### **WebSphere**

For information on how to configure WebSphere for HTTPS access, refer to the WebSphere documentation.

## **[Installing Host Publisher 3.5 using express logon on an existing installation of WebSphere Application Server 3.5](#)**

If you plan to use express logon and you are installing Host Publisher on an existing installation of WebSphere Application Server 3.5, the installation will complete successfully and both WebSphere Application Server and IBM HTTP Server (IHS) start properly. If, however, you then configure SSL with client authentication (required for express logon) through the IBM IHS Administration Server, the IHS loadmodule statements are incorrectly inserted in the `httpd.conf` file following the WebSphere loadmodule statements. This causes a syntax problem and results in a 1067 error.

To correct this problem, edit the `httpd.conf` file and move the WebSphere loadmodule statements so they appear after the IHS loadmodule statements.

## **[Appendix A. Performance and tuning](#)**

### **[System requirements](#)**

#### **[Host Publisher Studio](#)**

Host Publisher Studio requires:

- Intel platforms that support Microsoft Window 95, Windows 98, Windows NT, and Windows 2000

## Host Publisher Version 3.5 White Paper

- Pentium processor with a speed of 366 MHz or greater
- Minimum of 256 MB memory
- Minimum of 102 MB free disk space

Host Publisher Studio is a Java application and therefore makes heavy demands on the processor. If you are developing multiple complex applications, we suggest that you consider a more powerful processor.

### **Host Publisher Server**

Host Publisher Server is supported on Intel, RS/6000, Sun Sparc, IBM eSeries zServer, and iSeries hardware platforms, and performs best on modern machines that are designed to operate as servers. These machines are typically built for improved scalability and performance. Server machines usually have the capacity for large amounts of memory and offer large Level 1 (L1) and Level 2 (L2) caches.

The four major hardware components that most affect the capacity and performance of Host Publisher Server are:

- Central processing unit
- Memory
- Network interface card
- Hard drive

### **Central processing unit (CPU)**

Host Publisher runs as a plug-in to WebSphere Application Server and utilizes Host On-Demand to manage connections; Host Publisher, WebSphere Application Server, and Host On-Demand are all Java products. Typically, Java products perform best on fast or multiple processors. We recommend that Host Publisher run on modern processors that are designed for servers. These processors operate at Java-compatible speed and usually have built-in technologies, such as processor cache, that support greater capacity and performance.

Most modern, designed-for-server processors contain two levels of cache: L1 and L2. These caches act as temporary storage spaces for instructions and data obtained from slower memory.

For Intel-based servers, Pentium II or greater processors with speeds of 450 MHz or greater are recommended. Xeon processors provide enhanced performance over non-Xeon processors.

For IBM eServer zSeries, Turbo processors provide superior performance. We recommend the G5 and G6 models.

For iSeries servers, use a model that is recommended for Java applications. The *AS/400 V4R5 Performance Capabilities Reference* (found at

Host Publisher Version 3.5  
White Paper

<http://ca-web.rchland.ibm.com/perform/perfguideup/V4R5perfguide/V4R5perfguide.pdf>) lists models that are recommended for use with Java applications. For performance reasons, we recommend iSeries models with a processor commercial processing workload (CPW) rating of 460 or more. Models with lesser processor CPW ratings may work, but it is possible you will not be satisfied with your server performance. Refer to the most recent *AS/400 Performance Capabilities Reference* manual for the latest Java-recommended servers and their processor CPW ratings.

We recommend the iSeries models in the following table; however, all recently-announced servers might not be included here.

<b>iSeries Model</b>	<b>iSeries Feature</b>	<b>Processor CPW Rating</b>
820	2398	3200
	2397	2000
	2396	950
740	2070	4550
	2069	3660
730	2068	2890
	2067	2000
	2066	1050
	2065	560
720	2064	1600
	2063	810
270	2253	2000
	820	950
170	2388	1090
	2386	460
	2385	460
650	2243	2340
	2240	1794

Host Publisher Version 3.5  
White Paper

640	2239	998.6
	2238	583.3
530	2162	509.9
	2153	459.3
S40	2261	2340
	2256	1794
S30	2260	1794
	2259	998.6
	2258	583.3
S20	2166	759
53S	2157	509.9
	2156	459.3

## Memory

Another essential hardware resource is an adequate amount of physical memory; you must have enough to avoid memory depletion and excessive disk input/output.

By design, Java applications do not return their unused storage for reuse. The Java Virtual Machine (JVM) garbage collection facility runs only occasionally to claim unused storage. To avoid memory depletion between garbage collections, large amounts of memory are, therefore, required. In addition, you might want to tune your server for performance, which almost always affects memory allocation. Because of this, you cannot make changes to tuning parameters unless sufficient physical memory exists on the system.

We recommend that you allocate 512 MB of memory for Host Publisher running on uniprocessor machines, and 1 GB of memory for multiprocessor machines. These recommendations are in addition to what is required to run your other applications.

## Network interface card

The network interface card (NIC) can be an important factor in the capacity and performance of your server. The NIC moves data between the system data bus and the network media. Because the system data bus can operate at much higher speeds than the network media, the NIC can become a performance bottleneck. If the NIC is too slow, your server throughput/capacity will be based mainly on how fast the NIC can move data to the network media.

Choose a high-performance NIC that is designed for network-intensive applications. Such a NIC will implement one or more of these features:

- Bus mastering
- RAM buffering
- Direct memory access (DMA)
- Shared memory
- Onboard microprocessor

In addition, you might consider multiple NICs if you find that you saturate a single NIC and if your server has the capability to handle more workload.

### **Hard drive**

Your server must contain enough hard drive space to accommodate an operating system, a Java runtime environment (JRE), a Web server, WebSphere Application Server, Host Publisher, and any other products you plan to install. Your total disk space requirements should include extra space for configuration files, product upgrades, user applications files, and server operational files, such as error logs.

Actual disk space requirements are platform dependent. Host Publisher Server requires 200 MB of hard drive space. We recommend that you allocate at least 400 MB of hard drive space for Host Publisher Server, its operational files, and its user applications.

### **Hardware recommendations**

The following table depicts system hardware recommendations for each platform operating as a standalone Host Publisher server.

Machines with less power or fewer resources might work, but they might not allow you to reach your response time objectives and user capacity goals. Machines that are running other applications might require more resources, depending upon the resource consumption of those applications.

#### **Host Publisher Server Hardware Recommendations**

<b>Platform</b>	<b>Processor Type</b>	<b>Hard Disk Space</b>	<b>Memory</b>
<b>Intel</b>	Pentium 350 MHz or greater	2 GB	2 GB
<b>RS/600</b>	PowerPC 375 MHz or greater for uni-processor machines, PowerPC 332 MHz or greater for multiprocessor machines	2 GB	2 GB

Host Publisher Version 3.5  
White Paper

<b>Sun Sparc</b>	UltraSPARC 333 MHz or greater	2 GB	2 GB
<b>IBM eServer zSeries</b>	G4, G5, or G6 series	3 GB	2 GB
<b>iSeries</b>	Java-compatible with CPU rating of 260 or greater	2 GB	2 GB

### Server capacity

Many factors determine the number of users that your Host Publisher Server will support. These factors include, but are not limited to:

- The platform: computer hardware, operating system, and networking software
- The Web server: IBM HTTP Server, Microsoft IIS, and so forth
- The type of information served: 3270, 5250, database, VT, and so forth
- The application design:
  - Using connections with pooling enabled or disabled
  - Amount of interactions required between the server and the data source to obtain the requested information
  - Other similar factors
- Client usage patterns; for example, how often will users access the server?
- System performance objectives; for example, what are the response time requirements and what are the server availability requirements?
- Server workload from other applications

It is, therefore, not possible to provide capacities that are accurate in every system environment for all applications. For accurate capacities for your specific server, measure your specific applications in your operational environment.

When you estimate server capacity, remember to plan for growth and surges in user activity.

## [Appendix B: Manually Configuring WebSphere Application Server and your Web server for Host Publisher Server](#)

The following instructions are the WebSphere Application Server and Web server configuration steps required by Host Publisher. If Host Publisher Server installation fails or creates configuration problems, follow these instructions as a solution. If Host Publisher Server installation is successful, these steps are performed automatically during installation, requiring no additional configuration by you.

This information refers to Host Publisher's installation directory, which differs by installation and platform as follows:

- **For Windows NT**, <install\_dir> and <install\_dir2> is machine-specific. The default for both is C:\HostPub.
- **For AIX:**
  - <Install\_dir> = /usr/lpp/HostPublisher
  - <Install\_dir2> = /var/HostPublisher
- **For Solaris**
  - <Install\_dir> = /opt/HostPublisher
  - <Install\_dir2> = /var/HostPublisher
- **For OS/400**
  - <Install\_dir> = /QIBM/ProdData/HostPublisher
  - <Install\_dir2> = /QIBM/UserData/HostPublisher

### [Web Server Configuration](#)

Modify the Web server's configuration file to contain the following Host Publisher aliases:

#### **Running on WebSphere Application Server 3.5:**

```
Alias /HostPublisher /install_dir/Server/production/documents/  
Alias /_IBM_HP_WebAdmin_ /install_dir/Server/production/documents  
Alias /_IBM_HP_doc_ /install_dir/Common/doc/
```

**Note:** /HostPublisher/ is the default value. If you chose to replace /HostPublisher/ with something different during installation, use that value here.

The Web server's configuration file and location differs based on Web server software and installation choices. For example, the default location for the configuration file for IBM HTTP Server on Windows NT is c:\Program Files\IBM HTTP Server\conf\httpd.conf. Depending on the configuration you are modifying, you may have to use a Pass statement instead of an Alias

statement. Refer to other alias examples within the configuration to understand which statement to use. Do not change the aliases `_IBM_HP_WebAdmin_` and `_IBM_HP_doc_`.

## **Appendix C: Save classpath information from admin.config when installing WebSphere FixPacks**

**Note: If Host Publisher Server is already installed when you install a WebSphere Application Server FixPack, the FixPack installation recreates the WebSphere admin.config file, removing the Host Publisher classpath modifications from the file.**

To avoid the problem, you should save the portion of the admin.config classpath that was added by Host Publisher, for example, the entries that begin with the Host Publisher installation directory, into another file. After you install the FixPack, edit admin.config and add the Host Publisher classpath information back to the admin.config classpath. Then restart WebSphere Administrative Console. Note that for iSeries, the file that is equivalent to admin.config is called admin.properties.

The other alternative is to reinstall Host Publisher Server after installing the WebSphere FixPack.

## **Appendix D: Preliminary Check list and Software Installation Checklists for standard installations**

**Note: Not to be used for configuring more than one web server or using remote OSE**



# Host Publisher Version 3.5 White Paper

Step	Yes	No	Done	
				Preliminary checks before installing any software
1				Platform has sufficient memory (512K RAM Recommended)
2				Platform has sufficient disk space
3				Operating System software is at the correct level?
4				Operating System software has the correct PTFs or Fixpacks applied for this platform?
5				Web Server is on the list of supported web servers?
6				Web Server is has the correct FixPack or PTFs applied?
7				Is there a supported version of database software installed on the platform? DB2 6.1 Enterprise Edition for example
8				Is there a previous version of Host Publisher installed on this platform?
9				Is there a previous version of WebSphere Application Server installed on this platform?
				Software Installation Checklist for a totally new install of Host Publisher v3.5
1				Install a supported Web server on the machine
2				Install any PTFs or FixPacks for the Web server
3				Verify the Web Server is functioning properly before proceeding (run a test.html page from a browser and run admin tasks)
4				Install a supported Database product to be used by WebSphere AE 3.5 (DB2 6.1 EE for example)
5				Verify the Database software is installed correctly and is working before proceeding (can you access the SAMPLE db if using DB2)
6				Install WebSphere Application Server Advanced Edition (do not use the Quick Installation option which uses InstantDB)
7				Verify that WAS AE 3.5 has been successfully installed. (Service can be started)
8				Open the WAS AE 3.5 Administrator's Console (verify that it loads and the console is ready)
9				Start the default server from the WAS AE 3.5 Console (make sure it is "Running")
10				Open a browser and run Servlet Snoop to verify that WebSphere has installed successfully (http://Your_server/servlet/snoop)
11				Stop the Web server and WebSphere services in preparation for installing WAS 3.5 PTF 3
12				Install WebSphere AE 3.5 PTF 3 (follow the PTF 3 installation readme, the installation will inform you if it installed without errors)
13				Reboot the server and start the WebSphere service
14				Open a browser and run Servlet Snoop to verify that WebSphere PTF 3 has installed successfully
15				Install Host Publisher 3.5 Server software
16				Reboot the server
17				Verify that the Web server and WebSphere Services are started
18				Open the WAS AE 3.5 Administrator's Console (expand the topology view)
19				Verify that the HostPubServer Application Server is in the tree under the node
20				Highlight the HostPubServer and Start it. Verify that the state changes from "Stopped" to "Running"
1				Software Installation Checklist for upgrade from Host Publisher 2.1 or 2.2 to Host Publisher 3.5 on same platform
2				Verify that the platform that currently has Host Publisher 2.1 or 2.2 meets the Host Publisher 3.5 Hardware requirements
3				Verify that the platform software requirements meets the Host Publisher 3.5 Software requirements
4				If the platforms Hardware and Software requirements are met, then proceed to the next steps, if not perform the steps to upgrade the Hardware
5				Verify that a supported Database product is available for WebSphere AE 3.5 otherwise use DB2 EE v6.1 included with WAS AE 3.5
6				Install a Database product if needed and verify it has installed correctly and is working before proceeding
7				Is the Web server currently at a supported level for Host Publisher 3.5? If not install a supported Web server and apply required PTFs.
8				Verify that the Web server is functioning properly before proceeding (run test.html page from a browser and run admin tasks)
9				Upgrade the WebSphere Application Server from previous level to WAS 3.5 AE. (refer to WebSphere documentation to upgrade)
10				Verify that WAS AE 3.5 has been successfully installed (service can be started)
11				Open the WAS AE 3.5 Administrator's Console (verify that it loads and the console is ready)
12				Start the default server from the WAS AE 3.5 Console (make sure it is "Running")
13				Open a browser and run Servlet Snoop to verify that WebSphere has installed successfully (http://Your_server/servlet/snoop)
14				Stop the Web server and WebSphere services in preparation for installing WAS 3.5 PTF 3
15				Install WebSphere AE 3.5 PTF 3 (follow the PTF 3 installation readme, the installation will inform you if it installed without errors)
16				Reboot the server and start the WebSphere service
17				Open a browser and run Servlet Snoop to verify that WebSphere PTF 3 has installed successfully
18				Install Host Publisher 3.5 Server software into the same directory structure as the previous Host Publisher release
19				Reboot the server
20				Verify that the Web server and WebSphere Services are started
21				Open the WAS AE 3.5 Administrator's Console (expand the topology view)
22				Verify that the HostPubServer Application Server is in the tree under the node
23				Highlight the HostPubServer and Start it. Verify that the state changes from "Stopped" to "Running"

