

IBM WebSphere Host Access Transformation Services
Limited Edition



Getting Started

Version 5.0

IBM WebSphere Host Access Transformation Services
Limited Edition



Getting Started

Version 5.0

Note

Before using this information and the product it supports, be sure to read the general information under Appendix B, "Notices," on page 41.

2nd Edition (April, 2004)

© Copyright International Business Machines Corporation 2003, 2004. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Introducing HATS LE 1

Chapter 2. Migration 3

Migration from HATS LE Version 4. 3

Migration from HATS LE Version 5. 3

Chapter 3. Installing HATS LE 5

Prerequisites 5

Software requirements 5

Supported browsers and limitations 5

Installing HATS LE 6

Uninstalling HATS LE 6

Chapter 4. Configuring your application 9

The Welcome page 9

Configuring your host connection 9

Connection List 10

Using SSL security 11

Overriding session parameters in the URL . . . 12

Guaranteeing unique workstation IDs 12

Using a template to control your application's appearance 13

Using the keyboard and keypads 13

Advanced template configuration 14

Configuring your default transformation 14

Iterative configuration 15

Advanced configuration 15

Chapter 5. Administering your HATS LE Application 17

Viewing client host names in HATS LE

Administrative Console 18

Message logs and traces 18

Chapter 6. Modifying your application file 23

<application> tag 23

<connections> tag 23

<connection> tag 23

<defaultRendering> tag 23

<renderingSet> tag 23

<renderingItem> tag 24

<classSettings> tag 24

<class> tag 24

<setting> tag 25

<replace> tag. 27

Chapter 7. Modifying your connection file 29

<hodconnection> tag 29

<otherParameters> tag. 32

Appendix A. Remapping the keyboard in a HATS LE application 35

Concepts to understand before remapping HATS LE keys. 35

Determining keycode values. 36

Determining mnemonic keywords. 36

Example 37

Appendix B. Notices 41

Trademarks 42

Chapter 1. Introducing HATS LE

Welcome to IBM® WebSphere® Host Access Transformation Services Limited Edition (HATS LE) Version 5 (V5). With HATS LE you can quickly transform your 5250 host application into a Web application and make it available to your users through their browsers. HATS LE helps you protect your investment in legacy applications while presenting them to your end users—customers, suppliers, and internal staff—as Web applications accessible from their workstations.

HATS LE transforms the screens of a host application into Web pages that are part of a WebSphere application running on a Web server. End users access the pages from their Web browsers and use the pages to send and receive data from the host application. The end users might not even be aware that they are interacting with a host application; from their perspective, everything is taking place on the Web server.

Host Access Transformation Services Limited Edition is a functional subset of Host Access Transformation Services (HATS). HATS is a toolkit for IBM WebSphere Studio, and enables you to customize thoroughly the Web applications that transform your legacy host applications and make them available on your users' workstations. You can upgrade your HATS LE application to a HATS application, adding further customization and abilities to produce the function and ease of use that you want. HATS' rules-based transformation of host screens and support for iterative development mean you can add features as you are ready. With HATS you can:

- Extend 5250 and 3270 applications to Web browsers
- Deliver HTML directly to the desktop, requiring zero-footprint and zero-download
- Exploit the security and scalability of industry-leading WebSphere Application Server
- Combine information from multiple host screens into one Web page
- Incorporate your own business logic
- Provide legacy applications within enterprise portals through integration with WebSphere Portal
- Get on the Web quickly and customize at your own pace

For more information about HATS, visit the HATS Web site at <http://www.ibm.com/software/webservers/hats> or the HATS Information Center at <http://www.ibm.com/software/webservers/hats/library/version5/infocenter>

This document describes the functions of HATS LE and how to install and configure your HATS LE application. Before installing, be sure to review the *HATS LE Readme file*, which contains late-breaking information.

HATS LE offers simplified customization using an HTML Configuration Wizard and Administrative Console. You can install and configure HATS LE, and activate your application for use, in a very short time. Here are the steps you'll follow:

Migrate HATS LE:

If you currently use HATS LE V4 and you want to migrate your

application or if you want to migrate your HATS LE V5 application to HATS V5 Studio, follow the instructions in Chapter 2, “Migration,” on page 3.

Install HATS LE:

Review the prerequisites and install HATS LE on WebSphere Application Server. The instructions are in Chapter 3, “Installing HATS LE,” on page 5.

Configure your HATS LE application:

Follow the easy-to-use browser-based wizard to configure settings for your HATS LE application. Read all about it in Chapter 4, “Configuring your application,” on page 9.

Review your application:

The completion panel of the configuration wizard includes a link to start your application. Use this link to start your HATS LE application, log onto your host application, and review how the host screens are transformed. If you want to make improvements, restart the Configuration Wizard, make your changes, and review again until you are satisfied.

Manage your application:

After you have configured your HATS LE application, make it available to your users. Use the HATS LE Administrative Console to review the status and settings of your application and manage the connections to it. Learn more in Chapter 5, “Administering your HATS LE Application,” on page 17.

Chapter 2. Migration

If you currently use HATS LE V4 and you want to migrate your HATS LE V4 application to HATS LE V5, you must do so before upgrading to HATS LE V5.

If you upgrade without exporting your HATS LE V4 .zip file, you will lose your application. There is no auto-conversion from a HATS LE V4 application to a HATS LE V5 application.

Note: If you have HATS LE V4 and HATS LE V5 installed on the same machine, you will not be able to run both applications at the same time because they will share the same URL.

Migration from HATS LE Version 4

If you currently have HATS LE V4 installed, you will first need to export your application before you can migrate it into HATS LE V5. To export the HATS LE V4 application from the HATS LE Administrator Console into a .zip file:

1. Start the **HATS V4 LE Administrative Console** by entering the following URL in your Web browser: `http://<server_name>/HATSLE/admin` where `<server_name>` is the name of the server machine where your HATS V4 LE application runs.
2. Select **Advanced**
3. Select **Migrate**
4. Download and save as a .zip file

Once you have successfully exported your HATS LE V4 application, you will then need to install HATS LE V5. For more information, see Chapter 3, "Installing HATS LE," on page 5.

After you have upgraded, you will use the HATS LE V5 Administrative Console to upload and then migrate the .zip file that you packaged using the above steps.

Note: Your migrated templates and stylesheets will be redesignated with "_v4le" appended to the original name. Change the desired older templates to use the newer stylesheet style references before using in production. You can have both HATS LE V4 and HATS LE V5 installed on the same server at the same time, but you can only have one up and running at one time.

Migration from HATS LE Version 5

If you eventually want to move your application to the full HATS V5 Studio, you can export a HATS LE V5 application to a standard HATS V5 application:

1. Start the **HATS V5 LE Administrative Console** by entering the following URL in your Web browser: `http://<server_name>/HATSLE/admin` where `<server_name>` is the name of the server machine where your HATS V5 LE application runs.
2. Select **Advanced**
3. Select **Package**
4. Download and save as a .zip file

The generated .zip file (placed in a publicly accessible directory of the Limited Edition application) can then be downloaded and imported into HATS V5 Studio using the import wizard available in HATS Studio.

Note: This zip can only be imported into version 5 of HATS Studio.

Chapter 3. Installing HATS LE

This page lists the prerequisites for HATS LE and the steps for installing it.

Prerequisites

Software requirements

To install HATS LE, first make sure that you have installed the required software:

- iSeries™ Access Family, product ID 5722-XW1
- One of the prerequisite application servers. Table 1 shows the list of supported servers.

Table 1. HATS LE prerequisite servers

WebSphere Application Server	Version 5.02	Version 5.1
WebSphere Application Server (Base edition) for iSeries	✓	✓
WebSphere Application Server Network Deployment for iSeries	✓	✓
WebSphere Application Server-Express for iSeries	✓	✓

For information on the latest PTF level for each server, visit:

<http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver>

Supported browsers and limitations

Users interact with HATS LE applications through a browser. Table 2 shows the supported browser types.

Table 2. HATS LE supported browsers

Operating System	Netscape 6.2 or higher	Netscape 7.0 or higher	Internet Explorer 5.2.3 or higher	Internet Explorer 5.5 or higher	Mozilla 1.2.1 or higher	Opera 5.0 or higher	Konqueror 3.1 or higher	Safari 1.0 or higher
Windows®	✓	✓		✓	✓	✓		
AIX®		✓			✓			
Solaris		✓			✓			
Linux					✓	✓	✓	
Macintosh		✓	✓		✓	✓		✓

1. Bidirectional applications (such as Hebrew and Arabic) are not supported on Netscape browsers.
2. HATS LE Administration is only supported on Internet Explorer, Mozilla, and Netscape browsers.
3. With Opera, and Konqueror browsers, the function keys (F1 - F24) and the PageUp and PageDown keys on the keyboard do not perform any function. To enable the use of these keys in your applications, include a host keypad with these keys to enable the end user to send them to the host.

Note: Mozilla does support these functions although the behavior is different. The same behavior is also present when using a Netscape browser. The PageUp and PageDown functions will perform the action on the object selected (such as a drop-down list) first. Then, the actual action (PageUp or PageDown) will be sent to the host in the Mozilla/Netscape browser.

4. Konqueror browsers do not support retrieving the keyCode. Using the keyboard support, and pressing the Enter key with this browser might interact incorrectly with the host application. For applications that will run in Konqueror browsers, we recommend that you include a host keypad with the Enter key.
5. With the Safari browser, because of functional limitations, keyboard support is limited. For applications that will run in Safari browsers, we recommend that you include a host keypad with all the keys to enable the end user to send the keys to the host. For more information see Appendix A, "Remapping the keyboard in a HATS LE application," on page 35.

Please review the *HATS LE Readme file* for last-minute updates to prerequisites.

Installing HATS LE

Note: Users of WebSphere Application Server Network Deployment Version 5.02 or 5.1 for iSeries: it is not recommended to install HATS LE to any WebSphere instances that have been federated from WebSphere Application Server to WebSphere Application Server Network Deployment.

To install HATS LE on WebSphere Application Server Version 5.02 and 5.1 for iSeries, WebSphere Application Server Network Deployment Version 5.02 and 5.1 for iSeries, or WebSphere Application Server-Express for iSeries Version 5.02 or 5.1 follow these steps:

1. Start the WebSphere Administrative Console in your browser.
2. In the left column, expand **Applications** and click **Install New Application**.
3. **Browse** and find HATS5LE.ear from the local path.
4. Continue clicking **Next** until you reach the **Summary** screen. Click **Finish**.
5. Click the link to **Save to Master Configuration**.
6. Click **Save**. You will return to the main page of the WebSphere Administrative Console.
7. In the left column, expand **Applications** and click **Enterprise Applications**. In the right frame, click the check box next to HATS5LE.ear, and click **Start**.
8. In the left column, expand **Environment**, click on **Update Web Server Plugin** then Click **OK**.

Uninstalling HATS LE

To uninstall HATS LE on WebSphere Application Server Version 5.02 and 5.1 for iSeries, WebSphere Application Server Network Deployment Version 5.02 and 5.1 for iSeries, or WebSphere Application Server-Express for iSeries Version 5.02 or 5.1 follow these steps:

1. Start the WebSphere Administrative Console in your browser.
2. In the left column, expand **Applications** and click **Enterprise Applications**. In the list to the right, click the check box next to HATS5LE.ear, and click **Stop**.
3. Select the HATS5LE.ear again and click **Uninstall**. It will then be removed from the list.

4. Click the **Save** link above to go to the save page.
5. Click **Save**. You will return to the main page of the WebSphere Administrative Console.

Note: In some cases, log files (created while running your application with runtime tracing turned on) are not deleted when uninstalling your HATS LE application. To avoid this problem, before uninstalling your application, completely stop the application server instance where the HATS LE application is running. If these files still exist after uninstalling your HATS LE application, manually delete them (including the "HATS5LE.ear" directory under the "installedApps" directory) before reinstalling HATS LE.

Chapter 4. Configuring your application

After you install HATS LE, you can configure your HATS LE application from a supported browser on the local machine or on any other machine. The application server must be running when you configure your HATS LE application. Review the information in this chapter to see what information you need to supply to configure your HATS LE application. To start the HATS LE Configuration wizard, enter the following URL in your Web browser:

`http://<servername>/HATSLE/config`

where <servername> is the name of the server machine where you installed HATS LE. You must log in with a user name, which must have *SECADM authority, and password. You can specify the language in which you wish to use the Configuration wizard. As you use the wizard, click **Help** on any panel for more information.

The Configuration wizard can also be launched through the Administrative Console by selecting the **Configure** menu item on the left-hand side.

The Welcome page

When you log in to the Configuration wizard, you will see the welcome page. This page shows the status of your HATS LE application. If you have already configured and activated this application, there could be users connected to the application. These users will not be affected while you are reconfiguring your application. If users are connected to your HATS LE application when you click **Finish** to save your changes, they will remain connected to the host to which they connected when they started your HATS LE application. If you change connection settings, these users will use the new settings the next time they start your HATS LE application. If you make changes to the template or transformation settings, connected users will see the changes when their browser is refreshed.

This page also shows whether any other users are currently configuring this application. If you and another administrator configure your application at the same time, the administrator who saves configuration changes last will overwrite the changes saved by the other administrator.

Click **Continue** to proceed to the **Connection Settings** page.

Configuring your host connection

The Connection Settings page displays a table listing all of your connections. HATS LE allows for multiple connections giving you access to different hosts. Once the configuration has been made for the main connection and the first time configuration of the application is completed, you can configure multiple connections. After multiple connections are configured and active, you can choose which connection to use.

The table shows connection names, descriptions of the connections, host names as well as what connections are active to your application.

To make a connection active place a check in the **Active** box next to the desired connection in the table.

You can add additional connections by selecting the **Add** button, or edit an existing one by selecting **Edit**. Click **Remove** to delete a specific connection.

If you select the **Add** or **Edit** buttons, you will be asked for to supply the following information:

- A name for your connection
- A description of the connection
- The hostname or IP address of the host machine (default value of "localhost" is pre-filled)
- The port on which your HATS LE application will access the host machine (default value of "23" is pre-filled)
- The code page your host application uses
- How you want workstation IDs to be assigned
- Whether you want to use Secure Sockets Layer (SSL) security

Click **OK** when finished or **Cancel** to return to the **Connection Settings** page.

Connections can be placed in order of priority followed by descending order of importance. Use the **Move Up** or **Move Down** buttons to perform this operation. You can make a connection your default clicking the **Select Connection** radio button and then selecting the **Default** button. The default connection will then be highlighted in your connections settings table.

If you use a connection list, make sure you check the **Enable user connection list** check box. After you have successfully configured your application, a connection list will be added to the index page of your application. You can then select which connection to use from the list. For more information, see "Connection List."

Click **Finish** if you are done configuring your application, or **Next** to continue to the next step.

Note: The connection parameters can also be edited at any time by going to the Administrative Console and selecting the **Advanced** panel. See Chapter 7, "Modifying your connection file," on page 29 for more information.

Connection List

If you enabled the connection list by selecting the **Enable user connection list** check box on the **Connection Settings** page, you can modify how the connection list will look using the **Connection List Display Editor**.

To launch the editor, you must be in the **Administrative Console** and click on the **Advanced** link on the left-hand side. Click the **Edit** button to launch the editor.

Choose how you want your list to display by selecting either **Vertically, with the headers across the top** or **Horizontally, with the headers across the left side**.

The **Number of headers to display** allows you to pick up to seven header items:

- Connection
- Description
- Host name

- Code page
- Port
- SSL enabled
- Workstation ID

Any of these selections can be displayed as text, links or buttons.

If you check **Use translated header**, the text will display in the language selected for your application. If you do not check this option, you can enter the text you want to display.

To return to the **Advanced** page, click **Save** or **Cancel**.

Using SSL security

Using a secure connection over Secure Sockets Layer (SSL) encrypts data flowing over the connection and thus protects it against observation by a third party. If your HATS LE application transforms a 5250 application from a different machine, you can use SSL to protect the communications between your HATS LE application and the host machine.

For a connection to be secured, both the HATS LE application and the Telnet server on the host machine must support SSL. To secure the connection, the Telnet server must provide a certificate, which is used to encrypt the data. If the server does not have a certificate signed by a well-known certificate authority (Thawte, Verisign, or RSA), the HATS LE application must have a copy of the certificate to secure the connection.

If you want your HATS LE application to use SSL to communicate with the host, check **Enable SSL** on the Connection Settings panel. If a certificate is required by the HATS LE application, copy the required certificate from the Telnet server to the machine from which you are using the configuration wizard, and either select it from the drop-down list (if you have used it before) or click **Upload** to locate the file and copy it to the machine where your HATS LE application will run.

After configuring your application to use an SSL certificate (either the first time you configure the application or if you switch to a different certificate on a subsequent configuration), you will need to stop and restart your application. This is because SSL certificates are stored in a JAR file which may not get picked up by your application server without restarting your application.

Note: Depending on how your application server is configured for class reloading, your application may automatically restart when you change your SSL certificate. This is because the SSL certificate is stored in a JAR file that the application server is monitoring for changes. When a change occurs, the application server may restart your application to pick up this change. This process will disconnect any users currently using your application.

If you discover that after having selected an SSL certificate and clicked **Finish** in the Configuration wizard, the certificate is not being used when you access your application, follow these steps:

1. From the HATS LE Administrative Console, deactivate your application, so that no new users can access your application.
2. Stop/restart your application server instance.

3. Start the Configuration wizard and select the first item in the SSL certificate drop-down list (a blank item). Click **Finish**. This will cause the SSL certificate JAR to be deleted.
4. Restart the Configuration wizard and select the desired SSL certificate from the drop-down list.
5. Navigate through the wizard to the Summary page and check the option to reactivate your application. Click **Finish**.

Overriding session parameters in the URL

You can select which connection parameters can be overridden by your end user in the URL with which they access your HATS LE application.

To configure URL connection parameter overriding, edit the application.hap application settings file. You can edit this file from the **Advanced** section in the HATS LE Administrative Console. Edit the following section, under the <classSettings> section of the <application> section:

```
<class name="com.ibm.hats.common.DefaultConnectionOverrides">
  <setting name="prop1" value="true|false" />
  <setting name="prop2" value="true|false" />
</class>
```

Where "prop1" and "prop2" are the names of valid HOD session properties. Please refer to Modifying your application file for more information about the application settings file. All attribute names in the <session> tag of the application.hap file are also valid HOD property names, except for "workstationIDSource". For example, to allow your end users to override the host to which your HATS LE application connects, add the following:

```
<setting name="host" value="true" />
```

To allow your end users to override all connection parameters, set the following:

```
<setting name="allowAll" value="true" />
```

By default, parameter overriding is turned off for each connection parameter. You can explicitly allow all but one or more session parameters to be overridden by setting "allowAll" to "true" and then explicitly setting each parameter you do not wish to make overridable to "false". For example:

```
<class name="com.ibm.hats.common.DefaultConnectionOverrides">
<setting name="allowAll" value="true" />
<setting name="host" value="false" />
</class>
```

This configuration will allow your end users to override all connection parameters, except for the host name, via the URL.

If you configure your application to enable your users to override parameters, your end users can then supply these parameters in the initial request they make to your application. For example, to override the host and port number, assuming you have enabled overriding of these parameters, a user could enter this string into his browser:

```
http://<servername>/HATSLE/entry?host=mysecondary.host.com&port=992
```

Guaranteeing unique workstation IDs

If you choose to have HATS LE assign workstation ID values based on a string you supply, be sure to include an equal sign (=) before the wild-card character. For example, if you specify ABC* (without an equal sign) as the string from which

workstation IDs will be created, HATS LE cannot always assign unique workstation IDs. If you specify ABC=*, HATS LE will assign unique workstation IDs.

Using a template to control your application's appearance

Templates enable you to control the appearance of your HATS LE application. A template is a JSP file with an area reserved for the transformed host screen. The template can contain company logos, information, and links to other Web pages. A template also defines the background color for the area where the transformed host screen appears. Templates contain HTML and JSP code to include some or all of the following:

- At least one stylesheet (.css) file
- A banner area across the top of the page
- An area for the transformed host screen
- Host and application keypads

HATS LE supplies templates that you can use in your projects. You can see the names of these templates on the Template page of the configuration wizard. When you click the name of a template, a small preview appears on the right. Click **Preview** to see a larger preview.

Using the keyboard and keypads

Users frequently interact with host applications using special keys on the physical keyboard, such as F1, Attn, and Clear. There are two different ways in which the end users of your HATS projects can send keystrokes to the host:

- By pressing keys on the physical keyboard. If you want to enable your users to use these function keys on the keyboard, check Allow users to interact with the host using function keys on their keyboards. The keys will be passed directly to the host application.

Note: Make sure to check the HATS LE “Supported browsers and limitations” on page 5 for more information.

- By using a keypad, a graphical table of HTML buttons that represent keys on the physical keyboard. The end user clicks on the desired key in the keypad to send that host key to the host.

There are two keypads that you can add to your template:

- The host keypad, with keys (such as F1, Attn, and Clear) that represent host keys. These keys control functions on the host.
- The application keypad, with keys that represent application-level functions. These keys control functions within the HATS LE application. The application keypad keys include:

Reset Clears all the fields on the browser page of any entries made by the end user.

Reverse Screen

Toggles the screen image from a left-to-right image to a right-to-left image or vice-versa, if the application is running on a host with an Arabic or Hebrew code page. If an Arabic or Hebrew code page is not selected in the project connection settings, this button does not appear on the application keypad.

Refresh

Refreshes the current browser screen and performs the current transformation again.

Default

Turns off all configured settings and refreshes the current screen using the default settings. This can be useful if a user has trouble viewing a screen.

Disconnect

Disconnects from the host session. If this key is clicked, a link appears to let the user reconnect to the host.

Keyboard On/Off

Toggles support for using the physical keys on the host keyboard. If keyboard support is not enabled for the application, this button does not appear on the application keypad.

Note: The text of the button seen by the end user depends on the state of the keyboard. The button will read “Keyboard off” when keys are being sent to the host, and “Keyboard on” when keys are being sent to the browser.

By editing the application file, you can control which of these keys appear on the application keypad. For more information editing your application file, see Chapter 6, “Modifying your application file,” on page 23.

Advanced template configuration

You can perform more advanced configuration by adding a new template, editing an existing template, or exporting a template to your local system so that you can edit it with your favorite workstation editor. If you choose to export a template and edit, you can then add it to upload it to your HATS LE system. To edit this file, start the HATS LE Administrative Console and click **Advanced**. See Chapter 5, “Administering your HATS LE Application,” on page 17 for information about starting the HATS LE Administrative Console.

Configuring your default transformation

You can determine how your HATS LE application will transform components of the host screen into widgets on a Web page. For example, you can specify that selection lists should be transformed into drop-down lists or lists of links or buttons, how to recognize PF keys on your host screen, and how subfiles should be treated. You can say whether text copied from the host screen should stay the same color or have its color determined by the style sheet. Review the help for details on settings you can configure.

Default transformation features available for a HATS LE V5 application:

- Selection list rendering
- Detection of function keys on the host screen
- Subfile rendering
- Table detection

Iterative configuration

When you have finished configuring your HATS LE application, the Configuration wizard will provide a link to your application. Click this link to start the application, log in to your host system, and review how the host screens are transformed. If you want to make changes, you can restart the Configuration wizard and change the settings. When you reconfigure your HATS LE application, you might not need to revisit every panel in the Configuration wizard. For example, if you decide to change your template, you do not have to go back through the Transformation Settings panel. Whenever you have completed your changes, you can click **Finish** to save your changes and exit the wizard.

The first time you configure your HATS LE application, you will see an **Activate application** check box. By default this box is checked, so that your HATS LE application will be activated when you click **Finish**. The Configuration wizard remembers the status of this check box and uses it when you click **Finish** from an earlier panel. This means that if you have left the box checked, your application will be activated when you click **Finish** from any panel; if you cleared the check box, your application will not be activated when you click **Finish**. When you click **Finish**, your browser will show the completion page, which displays the URL for accessing your application.

Advanced configuration

The configuration wizard enables you to create a working Web application with a pleasing appearance. If you want to make further changes to your application, and you have HTML, XML, and JSP skills, you can edit your HATS LE application's file, connection file or template to perform further configurations.

To edit these files, start the HATS LE Administrative Console and click **Advanced**. See Chapter 5, "Administering your HATS LE Application," on page 17 for more information on the Administrative Console. See Chapter 6, "Modifying your application file," on page 23 for a description of the tags used in the application file and Chapter 7, "Modifying your connection file," on page 29 for a description of tags used in the connection file.

You can also make changes on how the connection list will look using the **Connection List Display Editor**. For more information, see "Connection List" on page 10.

Chapter 5. Administering your HATS LE Application

The HATS LE Administrative Console enables the system administrator to view sessions initiated for your HATS LE application. The application server must be running when you administer your HATS LE application.

To get to the HATS LE Administrative Console, enter the following URL in your Web browser:

`http://<servername>/HATSLE/admin`

where <servername> is the name of the server machine where your HATS LE application runs.

The Administrative Console home page shows the status of the application. You can enable or disable your application from this panel. You can also see the number of active connections. This is the default page when you enter the Administrative Console. Selecting any of the three menu items on the left-hand side of the Administrative Console window will open the appropriate page on the right-hand side:

Configure

The Configure section allows the administrator to specify settings for the application (using the Configuration wizard). The Configuration wizard is launched in the left-hand side of the Administrative Console window. It lists multiple configured connections in a table which can be sorted.

Manage Connections

This page displays a list of the active sessions connected to your HATS application and allows the administrator to manage all active connections to the application. You can sort the list by any column by clicking the column header. You can shut down a connection that has a problem.

Advanced

The Advanced section allows the administrator to configure more advanced settings of the application. This includes:

- Editing the application (.hap) or a connection (.hco) file
- Configuring the Connection List settings (see “Connection List” on page 10 for more information)
- Adding, editing, exporting, or deleting a selected template
- Packaging your application into a zip file which can then be imported into the HATS Studio (see Chapter 2, “Migration,” on page 3 for more information)

You can export your HATS LE application to a zip file that can be imported to HATS. Visit the HATS Web site at <http://www.ibm.com/software/webservers/hats> to find out what level of HATS is required for importing your HATS LE application.

Clicking **Home** at any time returns you to the **Administrative Console** page.

Viewing client host names in HATS LE Administrative Console

In the **Manage Sessions** section of the Administrative Console, you can see the IP address of each client that is connected to your HATS LE application (to view, click a session ID from the list of active sessions). In cases when clients connect through the IBM HTTP Server, as opposed to accessing your HATS LE application directly on your application server, the Session Details dialog might show only the IP address, not the fully resolved host name. To see the client host name, open the httpd.conf file in your IBM HTTP Server "conf" directory and add or edit the following: **HostnameLookups on**. You will need to stop and restart your HTTP server for this change to take affect. This will instruct the HTTP server to do a DNS lookup on the client IP address and determine its host name.

Note: Making this change will cause a slight performance degradation, because the HTTP server will have to resolve the IP address before passing the client's request to the application server.

Message logs and traces

When a HATS LE application runs in the WebSphere Application Server, logging and tracing is performed for the application. Logging and tracing is controlled by the settings of the runtime.properties file, in the root directory of HATS5LE.ear. In the drive and directory where you installed WebSphere Application Server, there is a folder named installedApps, which contains a folder named HATS5LE.ear.ear.

You can find your message log and trace files in the drive and directory where you installed WebSphere Application Server, They will be stored in the \HATS5LE.ear.ear\logs directory.

You can use a text editor to modify the runtime.properties file.

Note: The runtime.properties file must be edited with an ASCII editor. This can be done by transferring the file to a workstation (example: via FTP), editing the file and transferring it back.

The runtime.properties file contains the following basic properties:

maxTraceFiles

The maximum number of trace information files. The default is 5.

The base trace file name in runtime.properties is used as a template to generate unique sets of trace files for each application server. The default base name for a trace file is trace.txt, which can be changed.

An index (1, 2, 3, and so forth) is added to this name to distinguish multiple trace files. When trace1.txt reaches maxTraceFileSize, it is closed and renamed to trace2.txt. A new trace1.txt file is opened.

When the maxTraceFiles number is exceeded, the oldest file is deleted.

The trace*.txt files are created in the HATSLE directory.

maxTraceFileSize

Specifies the maximum size, in kilobytes, that a trace file reaches before an additional trace file is opened.

The value is a decimal integer. The default is 1024 KB.

traceFile

The name used as a template to generate file names for each set of application server files to which trace messages are written. The default base name for a trace file is trace.txt .

maxLogFiles

The maximum number of message files. The default is 2.

The base message log file name in runtime.properties is used as a template to generate unique sets of message log files for each application server. The default base name for a log file is messages.txt , which can be changed.

An index (1, 2, 3, and so forth) is added to this name to distinguish multiple message log files. When messages1.txt reaches maxLogFileSize, it is closed and renamed to messages2.txt . A new messages1.txt file is opened.

When the maxLogFiles number is exceeded, the oldest file is deleted.

The messages*.txt files are created in the HATSLE directory.

maxLogFileSize

Specifies the maximum size, in kilobytes, that a message log file reaches before an additional log file is opened.

The value is a decimal integer. The default is 512 KB.

logFile

The name used as a template to generate file names for each set of application server files to which log messages are written. The default base name for a trace file is messages.txt .

Note: Names of the properties are case-sensitive. Do not change the property name.

The runtime.properties file contains the following HATS LE application tracing properties:

trace.RUNTIME

Specifies the level of tracing for the main runtime and for all settings under RUNTIME.* that do not specify a trace level.

The value is an integer from 0-9. The default is 0.

See the description of the tracelevel.* keys for information on values for this setting.

trace.RUNTIME.WIDGET

Specifies the level of tracing for HATS widgets. This setting overrides the setting of trace.RUNTIME for tracing of widgets.

The value is an integer from 0-9. The default is 0.

See the description of the tracelevel.* keys for information on values for this setting.

trace.RUNTIME.ACTION

Specifies the level of tracing for HATS event actions. This setting overrides the setting of trace.RUNTIME for tracing of event actions.

The value is an integer from 0-9. The default is 0.

See the description of the `tracelevel.*` keys for information on values for this setting.

trace.RUNTIME.COMPONENT

Specifies the level of tracing for HATS components. This setting overrides the setting of `trace.RUNTIME` for tracing of components.

The value is an integer from 0-9. The default is 0.

See the description of the `tracelevel.*` keys for information on values for this setting.

trace.UTIL

Specifies the level of tracing for HATS runtime utilities.

The value is an integer from 0-9. The default is 0.

See the description of the `tracelevel.*` keys for information on values for this setting.

tracelevel.x

Each of the `tracelevel` keys specifies as its value a hexadecimal digit string. This string is a mask which is applied to the tracing feature for components which use that trace level. Each bit of the digit string controls one type of tracing for HATS.

The values of `tracelevel.1` through `tracelevel.7` should not be changed unless requested by IBM support. Otherwise, specifying these seven `tracelevel.*` properties is not necessary.

`Tracelevel.8` and `tracelevel.9` values can be used to create customized tracing levels.

The `runtime.properties` file contains the following Host On-Demand tracing properties:

trace.HOD.PS

Specifies the level of Host On-Demand presentation space tracing.

The value is an integer from 0-3. The default is 0.

trace.HOD.DS

Specifies the level of Host On-Demand data stream tracing.

The value is an integer from 0-3. The default is 0.

trace.HOD.TRANSPORT

Specifies the level of Host On-Demand transport tracing.

The value is an integer from 0-3. The default is 0.

trace.HOD.USERMACRO

Specifies the level of tracing for trace actions in Host On-Demand macros.

The value is an integer from 0-3. The default is 0.

trace.HOD.SESSION

Specifies the level of Host On-Demand session tracing.

The value is an integer from 0-3. The default is 0.

trace.HOD.PSEVENT

Specifies the level of Host On-Demand PS events.

The value is an integer from 0-1. The default is 0.

trace.HOD.OIAEVENT

Specifies the level of Host On-Demand OIA events.

The value is an integer from 0-1. The default is 0.

trace.HOD.COMMEVENT

Specifies the level of Host On-Demand COMM events.

The value is an integer from 0-1. The default is 0.

Note: You should not enable the Host On-Demand traces (except for PSEVENT, OIAEVENT, and COMMEVENT) unless requested by IBM Support.

Chapter 6. Modifying your application file

The application file contains XML tags that define the settings you select when you configure your HATS LE application. You can perform advanced configuration by editing the application file. To edit the file, start the HATS LE Administrative Console and click **Advanced**. This section describes some of the important tags which can be used in the application file along with their settings and classes.

<application> tag

The <application> tag is the enclosing tag for the application.

The attributes of the <application> tag are:

description

Contains an optional description for the application.

template

Specifies the name of the template for the application, which you select when you configure the application.

<connections> tag

The <connections> tag is the enclosing tag for the connection characteristics.

The attributes of the <connections> tag are:

default

Specifies the connection configured for the application. This value should always be main, and main is the default.

<connection> tag

The <connection> tag is the enclosing tag for the individual connection names.

The attributes of the <connection> tag are:

name Specifies the name of your connection.

Example:

```
<connections default="main">
  <connection name="main"/>
</connections>
```

<defaultRendering> tag

The <defaultRendering> tag is the enclosing tag for any default rendering settings you define in the project. This tag has no attributes.

<renderingSet> tag

The <renderingSet> tag is in rendering this the <defaultRendering> tag. Default is "main". HATS LE V5 only supports "main" as a valid value.

<renderingItem> tag

The <renderingItem> tag is the enclosing tag for a specific rendering item.

The attributes of the <renderingItem> tag are:

description

The description entered when the rendering item was created.

enabled

Indicates whether this rendering item is enabled. Reflects the state of the check box on the Rendering page of Project Settings.

endCol

The last column of the host screen to which this rendering item should be applied. -1 means the rightmost column of the host screen.

endRow

The last row of the host screen to which this rendering item should be applied. -1 means the bottom row of the host screen.

startCol

The first column of the host screen to which this rendering item should be applied.

startRow

The first row of the host screen to which this rendering item should be applied.

type The host component whose contents will be transformed.

widget

The widget into which the host component will be transformed.

<classSettings> tag

The <classSettings> tag is the enclosing tag for the Java™ classes you include in the application.

<class> tag

The <class> tag specifies the Java classes that can be included in an application.

The attributes of the <class> tag are:

name Specifies one of the following Java classes:

- com.ibm.hats.common.ApplicationKeypadTag
- com.ibm.hats.common.HostKeypadTag
- com.ibm.hats.common.KeyboardSupport
- com.ibm.hats.transform.components.*
where * is the name of a component for which you have customized a setting
- com.ibm.hats.transform.widgets.*
where * is the name of a widget for which you have customized a setting.
- com.ibm.hats.common.ClientLocale

The class names on the name attribute must be enclosed in quotes.

<setting> tag

The <setting> tag specifies the methods included in the Java class.

The attributes of the <setting> tag are:

name Specifies the name of the Java method or a customized setting for a component, a widget or the locale. The names listed depend on the Java class in which the methods reside or the name of a component or widget setting.

For the `com.ibm.hats.common.ApplicationKeypadTag` class, the methods are:

show If `value=true`, shows a keypad in the application.

style Depending on the value attribute, shows the keys defined with `value=true` as either a button or a link in the application keypad.

showKeyboardToggle

If `value=true`, shows a key in the application keypad for toggling display of a host keyboard.

showReset

If `value=true`, shows a Reset key in the application keypad to clear all the fields on the browser page of any entries made by the end user.

showReverse

If `value=true`, shows a Reverse key in the application keypad for bi-directional support.

showRefresh

If `value=true`, shows a Refresh key in the application keypad to refresh the browser window contents using the original transformation, and restore the input fields to their original value.

showDisconnect

If `value=true`, shows a Disconnect key in the application keypad to disconnect from the host.

showDefault

If `value=true`, shows a Default key in the application keypad to change the presentation to the default transformation.

For the `com.ibm.hats.common.HostKeypadTag` class, the methods are:

show If `value=true`, shows a host keypad in the application.

style Depending on the value attribute, shows the keys defined with `value=true` as either a button or a link in the host keypad.

showAttention

If `value=true`, shows an ATTN key in the host keypad.

showPrint

If `value=true`, shows a PRINT key in the host keypad for printing output.

showSystemRequest

If `value=true`, shows a SYSREQ key in the host keypad.

showClear

If `value=true`, shows a CLEAR key in the host keypad.

showFieldExit

If value=true, shows a Field exit key in the host keypad

showFieldPlus

If value=true, shows a Field+ key in the host keypad

showFieldMinus

If value=true, shows a Field- key in the host keypad

showPageUp

If value=true, shows a Page Up key in the host keypad.

showPageDown

If value=true, shows a Page Down key in the host keypad.

showPA1

If value=true, shows a PA1 key in the host keypad.

showPA2

If value=true, shows a PA2 key in the host keypad.

showPA3

If value=true, shows a PA3 key in the host keypad.

showReset

If value=true, shows a Reset key in the host keypad.

showEnter

If value=true, shows an Enter key in the host keypad.

showAltView

If value=true, shows an AltView key in the host keypad.

showHelp

If value=true, shows a Help key in the host keypad.

showF1 - showF24

If value=true, shows a Function key with a number in the host keypad.

For the com.ibm.hats.common.KeyboardSupport class, the methods are:

enable

Depending on the value attribute, enable specifies whether keyboard support is available in the application.

initialState

If value=true, the initial state of the host keyboard is on (the physical keys on the keyboard are active).

For the com.ibm.hats.component.* class or com.ibm.hats.widget.* class, name specifies a customized component or widget setting.

For the com.ibm.hats.common.ClientLocale class, name is always **locale**.

value For definitions of keypad keys, specifies whether to show the key in the keypad. Valid values are true and false.

For name=style, specifies how keys defined with value=true are displayed in the in the host keypad. Valid values are the following:

- Buttons
- Links

For component or widget settings, value specifies what you specified for the customized setting.

For the `com.ibm.hats.common.ClientLocale` class, value specifies characters that identify the country code of the locale.

<replace> tag

The <replace> tag specifies the text replacement values in a application.

The attributes of the <replace> tag are:

caseSensitive

Specifies whether the case of text replacement values must match before text replacement occurs. Valid values are true and false.

from Specifies the text you want to replace. The text on the from attribute must be enclosed in quotes.

to Specifies the text you want to insert in place of the value specified on the from attribute. The text on the to attribute must be enclosed in quotes.

Note: Care should be taken when using text replacement. Text replacement with a disparate number of characters in the strings can cause changes in the HTML representation of the screen. Depending on the widget used for presenting a region of a screen, text on a line of the screen could be contracted, expanded, or forced to a new line.

Chapter 7. Modifying your connection file

The connection file contains XML tags that define the settings you select when you configure your HATS LE connection. You can perform advanced configuration by editing the connection file. To edit the file, start the HATS LE Administrative Console and click **Advanced**. This section describes the tags used in the connection file.

<hodconnection> tag

The <hodconnection> tag specifies the connection characteristics for the application.

The attributes of the <hodconnection> tag are:

codePage

Specifies the numeric code page number for the code page used in the application. The default value is 037. You select the codePage value when you configure the application. A code page number might be used for more than one location or usage. See the description of the codePageKey attribute for the code page numbers.

codePageKey

Specifies the usage key that corresponds to the numeric codepage. The default value is KEY_US. Valid values for codePage and the location or usage key are:

Table 3. Code pages and usage keys

Code page	Usage key
037	KEY_BELGIUM KEY_BRAZIL KEY_CANADA KEY_NETHERLANDS KEY_PORTUGAL KEY_US
273	KEY_AUSTRIA KEY_GERMANY
274	KEY_BELGIUM_OLD
275	KEY_BRAZIL_OLD
277	KEY_DENMARK KEY_NORWAY
278	KEY_FINLAND KEY_SWEDEN
280	KEY_ITALY
284	KEY_SPAIN KEY_LATIN_AMERICA
285	KEY_UNITED_KINGDOM
290	KEY_JAPAN_KATAKANA_EX
297	KEY_FRANCE
420	KEY_ARABIC

Table 3. Code pages and usage keys (continued)

Code page	Usage key
424	KEY_HEBREW
500	KEY_MULTILINGUAL
803	KEY_HEBREW_OLD
838	KEY_THAI
870	KEY_BOSNIA_HERZEGOVINA KEY_CROATIA_KEY_CZECH KEY_HUNGARY KEY_POLAND KEY_ROMANIA KEY_SLOVAKIA KEY_SLOVENIA
871	KEY_ICELAND
875	KEY_GREECE
924	KEY_MULTILINGUAL_ISO_EURO
930	KEY_JAPAN_KATAKANA
933	KEY_KOREA_EX
937	KEY_ROC_EX
939	KEY_JAPAN_ENGLISH_EX
1025	KEY_BELARUS KEY_BULGARIA KEY_MACEDONIA KEY_RUSSIA KEY_SERBIA_MONTEGRO
1026	KEY_TURKEY
1047	KEY_OPEN_EDITION
1112	KEY_LATVIA KEY_LITHUANIA
1122	KEY_ESTONIA
1123	KEY_UKRAINE
1137	KEY_HINDI
1140	KEY_BELGIUM_EURO KEY_BRAZIL_EURO KEY_CANADA_EURO KEY_NETHERLANDS_EURO KEY_PORTUGAL_EURO KEY_US_EURO
1141	KEY_AUSTRIA_EURO KEY_GERMANY_EURO
1142	KEY_DENMARK_EURO KEY_NORWAY_EURO
1143	KEY_FINLAND_EURO KEY_SWEDEN_EURO
1144	KEY_ITALY_EURO
1145	KEY_LATIN_AMERICA_EURO KEY_SPAIN_EURO
1146	KEY_UNITED_KINGDOM_EURO

Table 3. Code pages and usage keys (continued)

Code page	Usage key
1147	KEY_FRANCE_EURO
1148	KEY_MULTILINGUAL_EURO
1149	KEY_ICELAND_EURO
1153	KEY_BOSNIA_HERZEGOVINA_EURO KEY_CROATIA_EURO KEY_CZECH_EURO KEY_HUNGARY_EURO KEY_POLAND_EURO KEY_ROMANIA_EURO KEY_SLOVAKIA_EURO KEY_SLOVENIA_EURO
1154	KEY_BELARUS_EURO KEY_BULGARIA_EURO KEY_MACEDONIA_EURO KEY_RUSSIA_EURO KEY_SERBIA_MONTEGRO_EURO
1155	KEY_TURKEY_EURO
1156	KEY_LATVIA_EURO KEY_LITHUANIA_EURO
1157	KEY_ESTONIA_EURO
1158	KEY_UKRAINE_EURO
1160	KEY_THAI_EURO
1364	KEY_KOREA_EURO
1371	KEY_ROC_EURO
1388	KEY_PRC_EX_GBK
1390	KEY_JAPAN_KATAKANA_EX_EURO
1399	KEY_JAPAN_ENGLISH_EX_EURO

delayInterval

Specifies the time (in milliseconds) that the server waits until a full host screen that is not the first host screen has arrived. The initial default value is 1200 milliseconds.

delayStart

Specifies the time (in milliseconds) that the server waits until the first full host screen has arrived. The initial default value is 2000 milliseconds.

description

Specifies a description for the connection configured for the application. This value is always empty.

SSL Specifies whether SSL is enabled. Valid values are:

true SSL is enabled for the application.

false SSL is not enabled for the application.

host Specifies the name of the host to which the application connects.

name Specifies the connection configured for the application. This value should always be main, and main is the initial default.

port Specifies the number of the port through which the connection to the host is made. The initial default is 23.

screenSize

Specifies the number of rows and columns that the host terminal displays. Valid values for screenSize are:

- 24 x 80
- 27 x 132

The initial default screen size is 24 x 80.

workstationID

Sets the workstationID property, which is used during enhanced negotiation for 5250. Values are in string format. If the value of workstationIDSource is "automatic" or "prompt", this value is ignored. If the value of workstationIDSource is "value", this value is used as the workstation ID. If the value of workstationIDSource is "session", this value specifies the name of the HTTP session variable from which the workstation ID is derived.

Note: If you select the **Use a specified value** option, you supply a string from which the workstation ID is created. You can also use wildcard characters in the specified string. The wildcard character is an asterisk (*). Include at least one wildcard character or only the first user will gain access to the application. Using the wildcard character increments the workstation ID using numbers, for example, WorkstationID1, WorkstationID2, and so on.

workstationIDSource

Specifies how the workstation ID value is set. Valid values are "automatic", "value", "prompt", and "session". The options are described in the help for the connection settings panel of the Configuration wizard.

<otherParameters> tag

The <otherParameters> tag specifies additional Host On-Demand session parameters.

Host On-Demand session parameters supported by HATS include:

Lamalef

Sets the LamAlef property, which determines whether LamAlef should be expanded or compressed. This property applies to Arabic sessions only. Values are in string format. Valid values are:

- LAMALEF_ON
- LAMALEF_OFF

The default value is LAMALEF_OFF.

numeralShape

Sets the numeralShape property. This property applies to bidirectional sessions only. Values are in string format. The default value is NOMINAL.

numericSwapEnabled

Sets the Numeric swapping property. This property applies to Arabic 3270 sessions only. Valid values are true and false. The default value is true.

roundTrip

Sets the roundTrip property. This property applies to bidirectional sessions only. Values are in string format. Valid values are:

- ROUNDTRIP_ON
- ROUNDTRIP_OFF

The default value is ROUNDTRIP_ON.

SecurityProtocol

Sets the SecurityProtocol property, which indicates whether to use the TLS v1.0 protocol or the SSL protocol for providing security. Values are in string format. The default value is TLS.

SSLServerAuthentication

Sets the SSLServerAuthentication property, which indicates whether SSL server authentication is enabled. Valid values are true and false. The default value is false.

symmetricSwapEnabled

Sets the Symetric swapping property. This property applies to Arabic 3270 sessions only. Valid values are true and false. The default value is true.

textOrientation

Sets the textOrientation property. This property applies to bidirectional sessions only. Values are in string format. Valid values are:

- LEFT_TO_RIGHT
- RIGHT_TO_LEFT

The default value is LEFT_TO_RIGHT.

ThaiDisplayMode

Sets the Thai display mode property. This property applies to Thai sessions only. Values are in string format. The default value is THAI_MODE_5.

workstationID

Sets the workstationID property, which is used during enhanced negotiation for 5250. Values are in string format. All lowercase characters are converted to uppercase. There is no default value.

Appendix A. Remapping the keyboard in a HATS LE application

Concepts to understand before remapping HATS LE keys

Every HATS LE application embeds code from the Host On-Demand product to handle the connection to the Telnet server on behalf of the web browser client. Therefore, this guide references Host On-Demand documentation for key mnemonics which may be sent to the host computer to represent the user pressing keys in the host session.

Note: Make sure to check the HATS LE “Supported browsers and limitations” on page 5 for more information. For Host On-Demand documentation, go to: <http://www.ibm.com/software/network/hostondemand/library>.

In every HATS LE application is a file called `KBS.js` which determines how the keycode values generated by the user’s keyboard are mapped to key signals sent to the host session. If it becomes necessary, you may edit the `KBS.js` file to remap the HATS LE keyboard. To locate `KBS.js`, look in the `HATS5LE.ear.ear/HATSLE.war/WebContent/Common` directory. There are three values in `KBS.js` which determine the relationship between any key on the keyboard being pressed and what signal is then sent to the host. These three values are:

1. The keycode value for the pressed key, represented as an integer variable in the `KBS.js` file. For example,

```
var CODE_F2 = 113;
```


for the ‘F2’ key,
2. The combination of the keycode value and the Alt, Ctrl, and Shift states of the keyboard, and
3. The HOD mnemonic keyword sent for the keycode combination value and Alt, Ctrl, and Shift states. For example,

```
[CODE_F2, 0, 0, 0, '[pf2]']
```

sends the pf2 key signal to the host when the F2 key is pressed on the user’s keyboard. To change key mappings in your HATS LE application, you can change or add combinations so new or existing keys send the mnemonic keyword. Note that to remap keys in HATS LE, you only need to modify the `keycode` and `defaultKeyMappings` variables at the top of the `KBS.js` file.

Note: There are dependencies between the keys listed in the `KBS.js` file, and the presence or absence of any key buttons listed on the screen. The rules governing these dependencies are listed below.

1. Keyboard Support must be enabled for keyboard keystrokes to have any effect in the HATS LE application. By default, HATS LE shows the application keypad but does not display the host keypad. In order for HATS LE to display the host keypad, you need to go to configure your `application.hap` file. For more information, see Chapter 6, “Modifying your application file,” on page 23.
2. The host keypad buttons and links displayed in a transformation determine which keyboard keys are enabled. If any host keypad buttons and links for host keys (those displayed by the Host Keypad, and those Host Keypad buttons you may add manually, if any) are displayed on a screen transformation, then only

those functions represented by the displayed keypad buttons and links have keyboard support. The Host Keypad buttons defined to display on a given transformation, including the default transformation, act as a filter to provide keyboard support to host keys mapped in KBS.js. Therefore, if you add a new mnemonic keyword to KBS.js while displaying any host keypad keys, you should also add a button or link for the new mnemonic keyword if you want to have keyboard support for that host key.

3. Keyboard support of host keys is not affected by transformations on host screen data and is only affected by which keypad buttons are displayed. This means that even if your transformation, including the default transformation, of the host screen data causes host key buttons and links to appear in the user's browser, only the host keys displayed in the Host Keypad or keypad buttons manually added to the transformation affect which host keys are supported from the keyboard.

Best practices:

With respect to the above rules when remapping the HATS LE keyboard, it is generally best to either:

- Not show the Host Keypad so all host keys defined in KBS.js are enabled on the keyboard by default, or
- Show the Host Keypad *and* add keypad buttons to the appropriate transformations for any additional host keys you add to KBS.js.

Determining keycode values

To determine the keycode value for a given key on the keyboard, you should refer to the your specific browser documentation

Determining mnemonic keywords

HATS LE-supported 5250 mnemonic keyword table

Table 4. Mnemonic Keywords for SendKeys

Function	Mnemonic Keyword
Attention	[attn]
Alternate View	[altview]
Clear	[clear]
Enter	[enter]
F1	[pf1]
F2	[pf2]
F3	[pf3]
F4	[pf4]
F5	[pf5]
F6	[pf6]
F7	[pf7]
F8	[pf8]
F9	[pf9]
F10	[pf10]
F11	[pf11]

Table 4. Mnemonic Keywords for SendKeys (continued)

Function	Mnemonic Keyword
F12	[pf12]
F13	[pf13]
F14	[pf14]
F15	[pf15]
F16	[pf16]
F17	[pf17]
F18	[pf18]
F19	[pf19]
F20	[pf20]
F21	[pf21]
F22	[pf22]
F23	[pf23]
F24	[pf24]
Field exit	[fldext]
Field+	[field+]
Field-	[field-]
Help	[help]
PA1	[home]
PA2	[pa1]
PA3	[pa2]
System Request	[pa3]
Page Up	[pageup]
Page Down	[pagedn]
Reset	[reset]
System Request	[sysreq]
Test Request	[test]

Example

The following is an example on how to map the Ctrl+Home key combination on the keyboard. To send the Home key to the host session and also place a button for this function in your transformation, two steps are required: Edit KBS.js and then modify your transformation as shown below.

Specifying the key combination that sends the host key

Modify the KBS.js file by adding the lines specified in ***bold italic*** below:

```
var CODE_BACKSPACE = 8;
var CODE_TAB       = 9;
var CODE_ENTER     = 13;
var CODE_PAUSE     = 19;
var CODE_ESC       = 27;
var CODE_PAGEUP    = 33;
var CODE_PAGEDOWN  = 34;
var CODE_END       = 35;
```

```

var CODE_HOME      = 36;
var CODE_INSERT    = 45;
var CODE_DELETE    = 46;
var CODE_A         = 65;
var CODE_B         = 66;
var CODE_C         = 67;
var CODE_D         = 68;
var CODE_E         = 69;
var CODE_F         = 70;
var CODE_G         = 71;
var CODE_H         = 72;
var CODE_I         = 73;
var CODE_J         = 74;
var CODE_K         = 75;
var CODE_L         = 76;
var CODE_M         = 77;
var CODE_N         = 78;
var CODE_O         = 79;
var CODE_P         = 80;
var CODE_Q         = 81;
var CODE_R         = 82;
var CODE_S         = 83;
var CODE_T         = 84;
var CODE_U         = 85;
var CODE_V         = 86;
var CODE_W         = 87;
var CODE_X         = 88;
var CODE_Y         = 89;
var CODE_Z         = 90;
var CODE_HOSTHOME = 36;
var CODE_F1        = 112;
var CODE_F2        = 113;
var CODE_F3        = 114;
var CODE_F4        = 115;
var CODE_F5        = 116;
var CODE_F6        = 117;
var CODE_F7        = 118;
var CODE_F8        = 119;
var CODE_F9        = 120;
var CODE_F10       = 121;
var CODE_F11       = 122;
var CODE_F12       = 123;
var HostKey        = 1;
var ApplicationKey  = 2;
/*****
* NOTICE: DO NOT MODIFY THE ABOVE VARIABLES!!
*****/

var defaultKeyMappings = [
// KEYCODE,      ALT, CTRL, SHIFT, MNEMONIC
//===== command key mappings =====
[CODE_ENTER,     0,    0,    0, '[enter]'      ],
[CODE_PAUSE,     0,    0,    0, '[attn]'       ],
[CODE_ESC,       0,    0,    0, '[clear]'      ],
[CODE_ESC,       0,    0,    1, '[sysreq]'     ],
[CODE_PAGEUP,    0,    0,    0, '[pageup]'     ],
[CODE_PAGEUP,    1,    0,    0, '[refresh]'    ],
[CODE_PAGEDOWN,  0,    0,    0, '[pagedn]'     ],
[CODE_HOSTHOME,  0,    1,    0, '[home]'       ],
[CODE_PAGEDOWN,  1,    0,    0, '[pa3]'        ],
[CODE_END,       1,    0,    0, '[pa2]'        ],
[CODE_INSERT,    1,    0,    0, '[default]'    ],
[CODE_DELETE,    1,    0,    0, '[pa1]'        ],
[CODE_D,         0,    1,    0, '[disconnect]' ],
[CODE_H,         0,    1,    0, '[help]'       ],
[CODE_P,         0,    1,    0, '[printhost]'  ],
[CODE_J,         0,    1,    0, '[printjobs]'  ],

```

```

[CODE_ENTER,      1,  0,  0, 'reverse' ],
[CODE_K,          0,  1,  0, 'toggle'  ],
[CODE_S,          0,  1,  0, 'ResetButton' ],
//===== function key mappings =====
[CODE_F1,         0,  0,  0, '[pf1]'   ],
[CODE_F1,         0,  0,  1, '[pf13]'  ],
[CODE_F2,         0,  0,  0, '[pf2]'   ],
[CODE_F2,         0,  0,  1, '[pf14]'  ],
[CODE_F3,         0,  0,  0, '[pf3]'   ],
[CODE_F3,         0,  0,  1, '[pf15]'  ],
[CODE_F4,         0,  0,  0, '[pf4]'   ],
[CODE_F4,         0,  0,  1, '[pf16]'  ],
[CODE_F5,         0,  0,  0, '[pf5]'   ],
[CODE_F5,         0,  0,  1, '[pf17]'  ],
[CODE_F6,         0,  0,  0, '[pf6]'   ],
[CODE_F6,         0,  0,  1, '[pf18]'  ],
[CODE_F7,         0,  0,  0, '[pf7]'   ],
[CODE_F7,         0,  0,  1, '[pf19]'  ],
[CODE_F8,         0,  0,  0, '[pf8]'   ],
[CODE_F8,         0,  0,  1, '[pf20]'  ],
[CODE_F9,         0,  0,  0, '[pf9]'   ],
[CODE_F9,         0,  0,  1, '[pf21]'  ],
[CODE_F10,        0,  0,  0, '[pf10]'  ],
[CODE_F10,        0,  0,  1, '[pf22]'  ],
[CODE_F11,        0,  0,  0, '[pf11]'  ],
[CODE_F11,        0,  0,  1, '[pf23]'  ],
[CODE_F12,        0,  0,  0, '[pf12]'  ],
[CODE_F12,        0,  0,  1, '[pf24]'  ]
];

```

Note: The name given to any new keycode variable (CODE_HOSTHOME in this example) is not important except that it be unique and match the corresponding entry in the defaultKeyMappings variable list. Also note in this example that adding a new keycode variable for the Home key is not actually necessary since the

```
var CODE_HOME = 36;
```

entry already exists for the Home key's keycode. However, if you are mapping a keyboard key that is not associated with a keycode variable already defined in KBS.js, such an entry would be necessary.

Due to limitations in how the Macintosh Safari browser maps keys to ordinal values, we have the need to remove the Function key definitions from the default key mappings. If the Macintosh Safari browser is detected in keyboard init, the table defined below will be replaced by the default table defined above.

```

//*****
var macSafariKeyMappings = [
// KEYCODE,      ALT, CTRL, SHIFT, MNEMONIC
//===== command key mappings =====
[CODE_ENTER,    0,  0,  0, '[enter]' ],
[CODE_PAUSE,    0,  0,  0, '[attn]'  ],
[CODE_ESC,      0,  0,  0, '[clear]' ],
[CODE_ESC,      0,  0,  1, '[sysreq]' ],
[CODE_R,        0,  1,  0, '[reset]' ],
[CODE_PAGEUP,   1,  0,  0, '[refresh]' ],
[CODE_PAGEDOWN, 0,  0,  0, '[pagedn]' ],
[CODE_PAGEDOWN, 1,  0,  0, '[pa3]'   ],
[CODE_END,      1,  0,  0, '[pa2]'   ],
[CODE_INSERT,   1,  0,  0, '[default]' ],
[CODE_DELETE,   1,  0,  0, '[pa1]'   ],
[CODE_D,        0,  1,  0, '[disconnect]' ],
[CODE_H,        0,  1,  0, '[help]'  ],
[CODE_P,        0,  1,  0, '[printhost]' ],

```

```

[CODE_J,      0,    1,    0, 'printjobs' ],
[CODE_ENTER,  1,    0,    0, 'reverse'   ],
[CODE_K,      0,    1,    0, 'toggle'   ],
[CODE_S,      0,    1,    0, 'ResetButton' ],
[CODE_ENTER,  0,    1,    0, '[fldext]' ],
];

```

Note: The Function keys are not mappable in Safari.

Appendix B. 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 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 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
TL3B/062
3039 Cornwallis Road
RTP, 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 information 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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- IBM
- iSeries
- WebSphere

Pentium is a trademark of Intel Corporation in the United States, other countries, or both. (For a complete list of Intel trademarks see <http://www.intel.com/sites/corporate/tradmarx.htm>.)

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. SPARC is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Netscape is a registered trademark of Netscape Communications Corporation in the United States and other countries.

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



Readers' Comments — We'd Like to Hear from You

IBM WebSphere Host Access Transformation Services Limited Edition
Getting Started
Version 5.0

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Host Access Information Development
Department E40D/Building 502 Research
Triangle Park, NC 27709-9990



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Printed in USA