



# End-to-End: IBM WebSphere Host Access Transformation Services (HATS) demonstration on the IBM eServer iSeries platform

by Michael J. Sandberg  
IBM eServer Solutions Enablement  
August 2005



## Table of contents

<b>Abstract</b> .....	<b>1</b>
<b>Introduction</b> .....	<b>1</b>
<b>Prerequisites</b> .....	<b>2</b>
<b>Creating and reviewing a default HATS project</b> .....	<b>3</b>
<b>Creating a HATS project</b> .....	<b>3</b>
Running the HATS project in the test environment .....	3
<b>Reviewing FLGHT400 in HATS</b> .....	<b>4</b>
<i>Starting the application</i> .....	5
<i>Creating a new reservation</i> .....	7
<b>Customizing the HATS project</b> .....	<b>13</b>
Understanding the HATS features .....	13
<i>Extend host applications to the Web quickly</i> .....	13
<i>Transform host screen components in real time</i> .....	13
<i>Provide programmed navigation through multiple legacy screens</i> .....	13
<i><b>not defined.</b></i>	
<i>Integrate with WebSphere software</i> .....	13
Customizing FLGHT400.....	13
<i>Modifying default renderings</i> .....	13
<i>Adding a calendar pop-up</i> .....	18
<b>Summary</b> .....	<b>25</b>
<b>Additional information</b> .....	<b>26</b>
<b>About the author</b> .....	<b>26</b>
<b>Trademarks and special notices</b> .....	<b>27</b>

## Abstract

*This online lab provides a demonstration of using IBM® WebSphere® Host Access Transformation Services (HATS) to deliver a better user experience for green-screen applications that run on the IBM eServer™ iSeries™ platform. This demonstration uses the IBM FLGHT400 sample application, which has been written as a typical example of the RPG III monolithic, server-controlled application model that existed early in the life of the iSeries predecessor family of servers (IBM AS/400® servers). This lab is for RPG programmers who are beginners in the world of graphical and Web-based programming techniques.*

## Introduction

Traditionally, the FLGHT400 application has only been accessible through a green-screen interface. This leads to the perception that it is not a modern application and that the user either needs to be directly attached to the server via a terminal or running emulation software on a client. These two problems are addressed by the IBM WebSphere Host Access Transformation Services (HATS) product. This enables the FLGHT400 application interface to be displayed in a browser, giving it a better user interface (UI) and eliminating the need to have any code that must be loaded and maintained on the client except a Web browser. In addition to a simple presentation of FLGHT400 in a browser, HATS allows a great deal of customization to the look, feel, and function of the application.

This lab follows the same general steps as in the [FLGHT400 Overview](#) article to demonstrate how HATS enhances the user experience of the FLGHT400 application. Throughout this lab, embedded URLs (in blue) link to external Web sites. These links are also available in the “Additional information” section of this document.

## Understanding the HATS features

- HATS can facilitate FLGHT400 in many ways to become more user friendly and functional.

The following are some of the HATS features that can be used to extend the FLGHT400 application:

**Extend host applications to the Web quickly**

- The HATS rules-based transformation engine makes it possible to extend host applications to the Web within hours of installing the software. HATS is a zero-footprint, zero-download, Web-to-host solution. As mentioned, the only software needed on the client is a Web browser.

**Transform host screen components in real time**

- The power of HATS lies in its ability to accurately recognize and transform the components of host screens in real time to a Web interface according to a set of predefined rules. It is easy to modify the rules to accommodate the specific needs of the application. With HATS, a variety of elements can be added to host screens, such as: drop-down lists, hot links, tables, buttons, valid value lists, tabbed folders, and graphs. HTML elements can also be added, such as: logos, graphics, backgrounds, and Web links.

**Provide programmed navigation through multiple legacy screens**

- HATS macro support allows programmed navigation through multiple legacy screens to improve the productivity and ease-of-use of host applications. HATS enables programmed access to a single host application. It also integrates screens from multiple host applications into a single Web interface. Macros created in IBM WebSphere Host On-Demand can also be used.

**Integrate with WebSphere software**

The HATS Toolkit is fully integrated with the Eclipse-based IBM Rational software development platform. It offers an intuitive interface for customizing the rules for transformation of host screens. The HATS applications can be deployed to both WebSphere Application Server and WebSphere Portal, and can take advantage of the extensive security and reliability features found in both platforms.

## Prerequisites

To complete the steps detailed in this paper, load and configure a development environment for HATS using one of the following methods:

- It is recommended that solution providers use the [IBM Software Access Catalog](#) to download and install WebSphere Development Studio Client Advanced Edition for iSeries V6.0 and Host Access Transformation Services V6.
- Alternatively, it is possible to download a no-cost trial development environment version of IBM [Rational® Web Developer for WebSphere Software V6.0](#) and a trial version of [WebSphere Host Access Transformation Services Toolkit](#).

Details on prerequisites and installation of HATS can be found in the [IBM WebSphere Host Access Transformation Services \(HATS\) V6 Information Center](#).

## Creating and reviewing a default HATS project

There are several, easy-to-follow processes to create and then review graphical user interfaces for traditional RPG III programs. The defaults make it even easier, though the defaults can be changed as needed.

### Creating a HATS project

To create a default HATS project, follow these steps:

1. Launch **HATS Studio 6**.
2. Click on **launch the Create HATS Project wizard**.
3. Enter a name for the project. Then click **Next**.
4. Enter the host name of the iSeries server and the port (if it is not port 23).
5. Select **5250** from the **Type:** drop-down window.
6. Review the other default values and adjust as needed.
7. Click **Finish**.

After these steps are complete, a default HATS project will be created.

### Running the HATS project in the test environment

When the HATS project is created, it can be previewed live at any time by using the **Run on Server** command. This section describes that process and walks through the FLGHT400 application as it would appear when running with HATS using the following steps:

1. Right-click the **HATS Project** and select **Run on Server**. This will launch the test environment (Figure 1).
2. To see the host view while running the test environment, click **Yes**; otherwise click **No**.
3. Select the text environment and check the **Set server as the project default** option. This allows the HATS project to run in the future on the same test server without being prompted again.
4. Click **Finish**.

## End-to-End: IBM WebSphere Host Access Transformation Services (HATS) demonstration on the IBM eServer iSeries platform

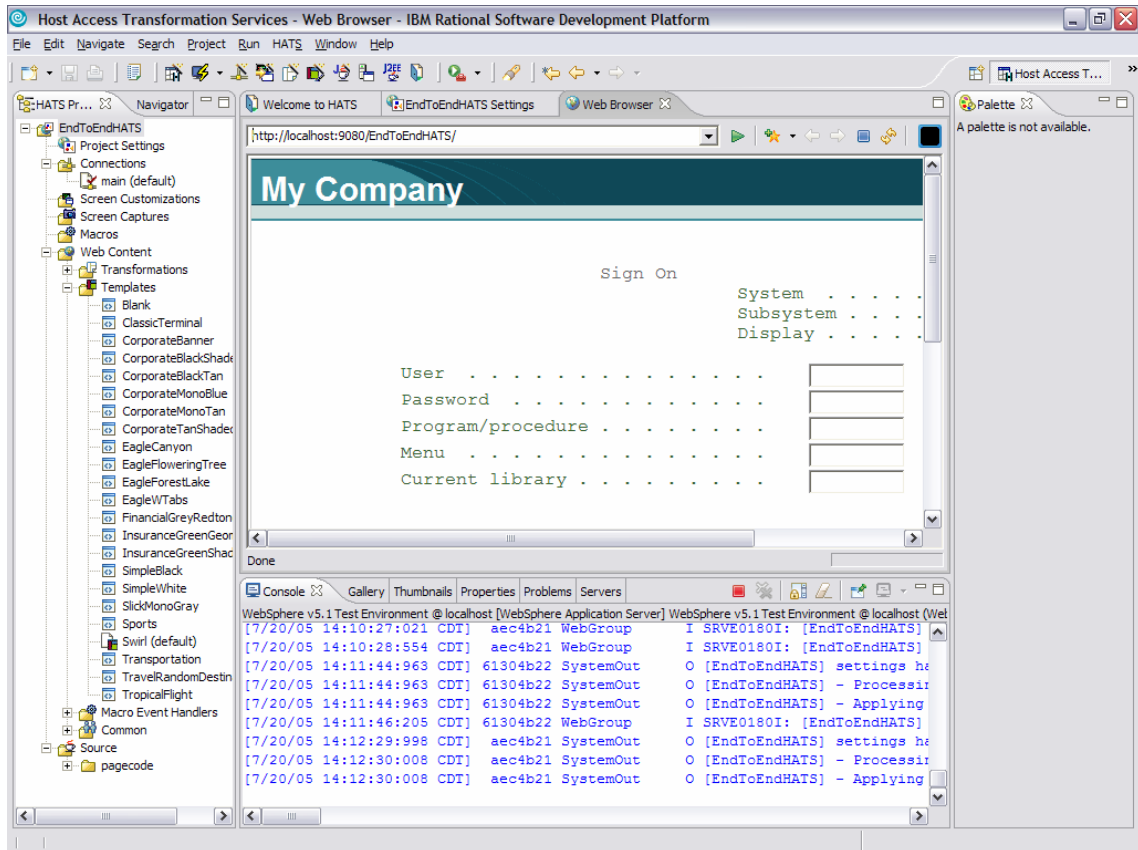


Figure 1: Test environment

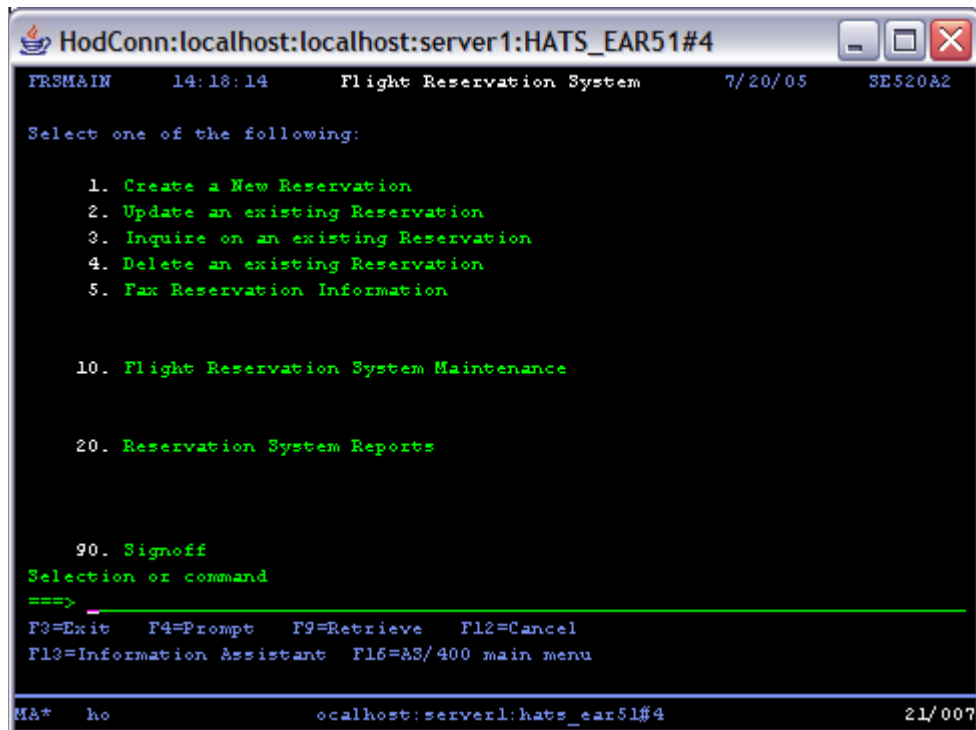
### Reviewing FLGHT400 in HATS

Now that a default HATS project has been created and can run in the test environment using **Run on Server**, it is time to review FLGHT400 using HATS. The same steps will be followed as in the [FLGHT400 Overview](#) article. Notice on some of the field prompts that HATS has trouble rendering a few of the subfiles. This will be corrected in the **Customize FLGHT400** section later in this paper.

## Starting the application

Follow these steps to start the FLGHT400 application:

1. Log on to the iSeries system in the HATS Web browser just launched. The FLGHT400 application is started by running the following command:  
**Go frsmain**
2. When the **Flight Reservation System** application starts, it will display the green-screen version of the Flight Reservation System application as shown in Figure 2.



```
HodConn:localhost:localhost:server1:HATS_EAR51#4
-----
FRSMAIN      14:18:14      Flight Reservation System      7/20/05      SE520&2

Select one of the following:

    1. Create a New Reservation
    2. Update an existing Reservation
    3. Inquire on an existing Reservation
    4. Delete an existing Reservation
    5. Fax Reservation Information

    10. Flight Reservation System Maintenance

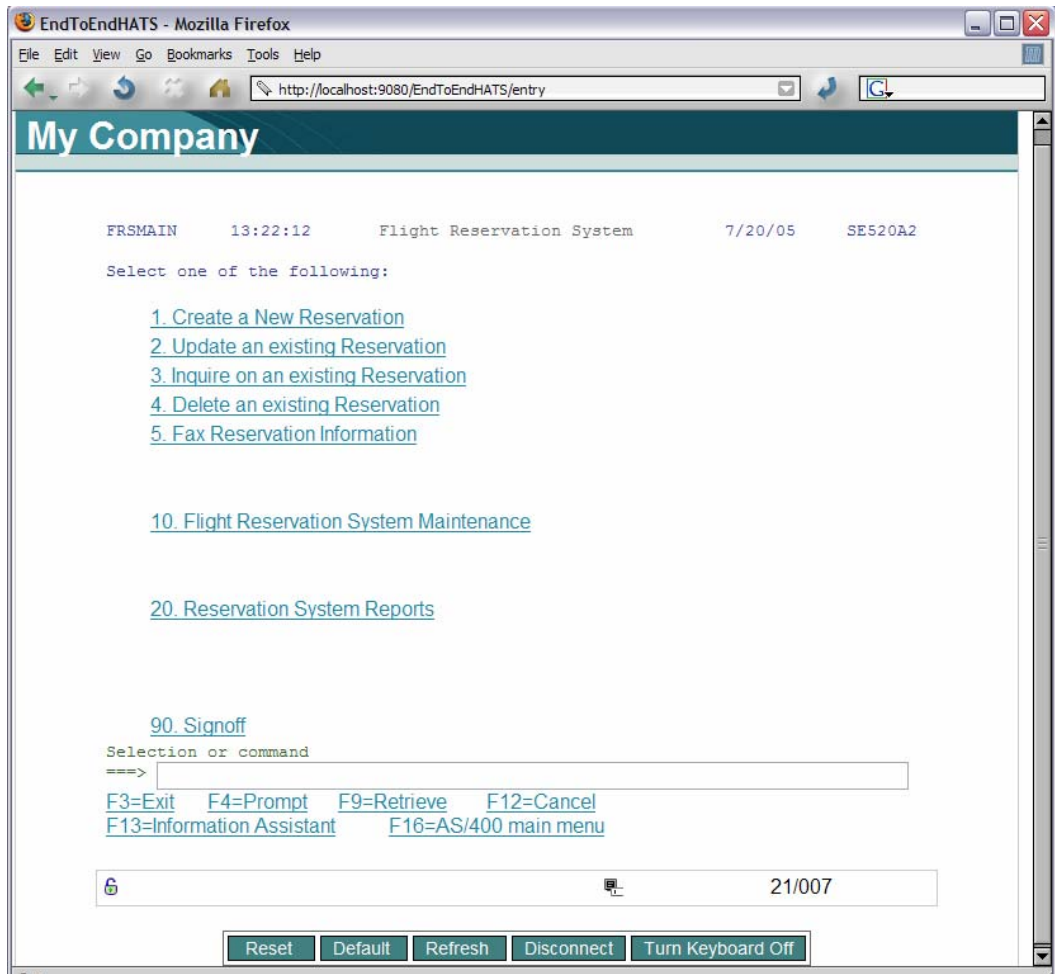
    20. Reservation System Reports

    90. Signoff
Selection or command
===>
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
F13=Information Assistant   F15=88/400 main menu

HA*   ho                               ocalhost:server1:hats_ear51#4      21/007
```

Figure 2: Flight Reservation System application

3. Figure 3 is the default HATS rendering of the **Flight Reservation System** application screen using the **Swirl** template. Notice how the menu and function keys have been transformed into clickable links even though the command line access is still available. The basic look can easily be altered by changing the template used and modifying the default rendering options for the project.



**Figure 3: Default HATS rendering of the Flight Reservation System application**

4. To change the default template, expand the HATS project folder.
5. Expand the **Web Content** folder and the **Templates** folder.
6. Right-click any template and select **Set as Default Template**.



Figure 4 is an example of the **CorporateMonoTan** template.

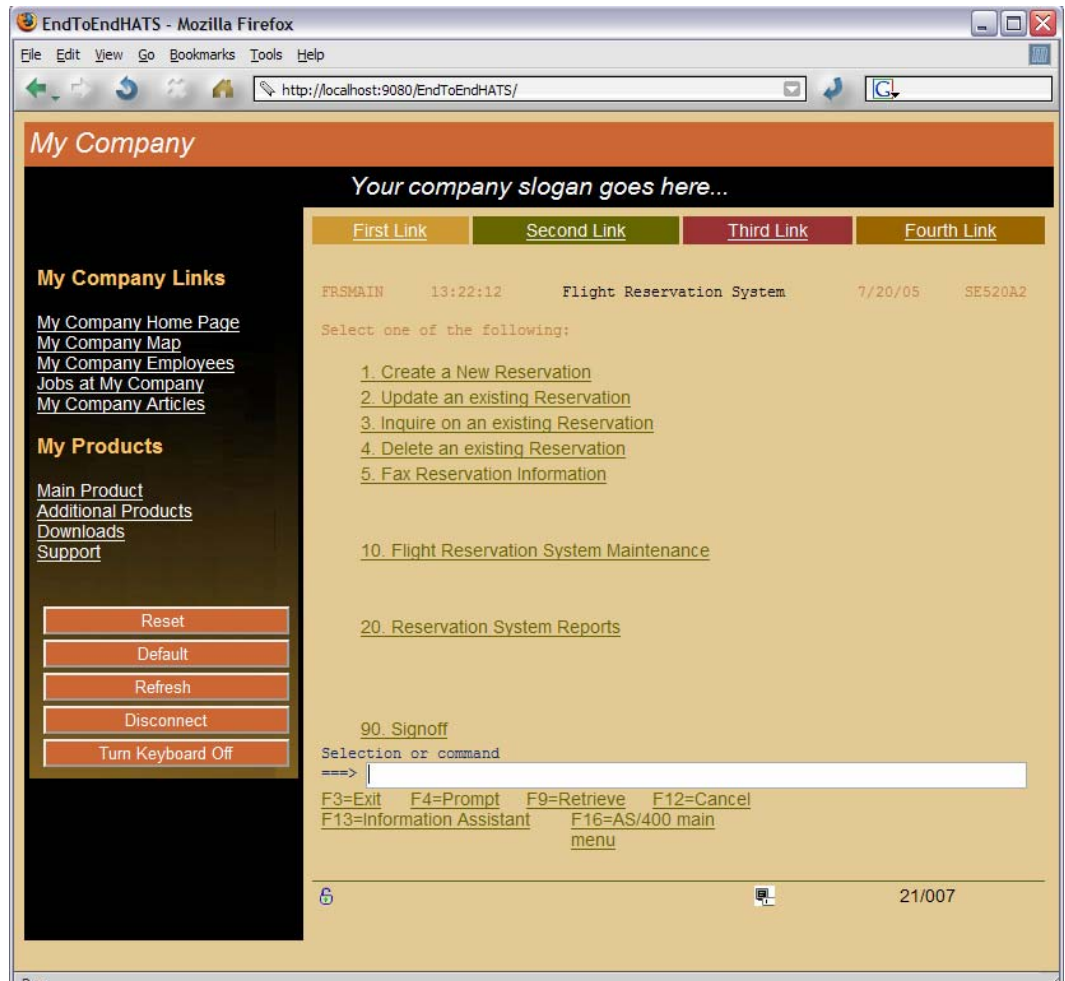


Figure 4: Example of the **CorporateMonoTan** template

### Creating a new reservation

One menu item explained in this scenario is **Create a New Reservation** (refer to Figure 4):

1. Select the **1.Create a New Reservation** option from the main menu. This calls the RPG program FRS001. This program uses FRS001DF for display files and reads and writes its data to the database files: FRCITY, TOCITY, CUSTNAME, CUSTOMER, FLIGHTSZ, and ORDERS.
2. From the **FRS001DF** display screen, press the **F4** function key. This calls the RPG program FRS402. (**Note:** When the **F5** function key is pressed, it calls FRS403, and so on. These programs, in turn, use the display files and database tables.)

3. The application will now prompt for the **Agent Name** and **Password** (as shown in figures 5 and 6). Use any of the agent names and passwords listed in the **Agents** table in the *FLGHT400 Library* and press **F10** to log on.

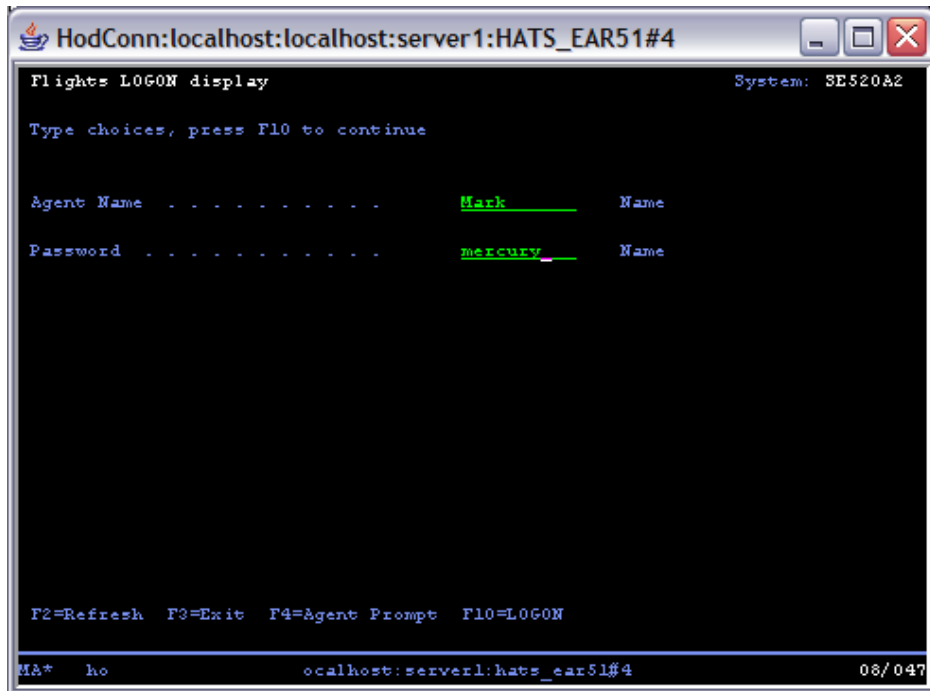


Figure 5: Agent Name and Password green-screen rendering

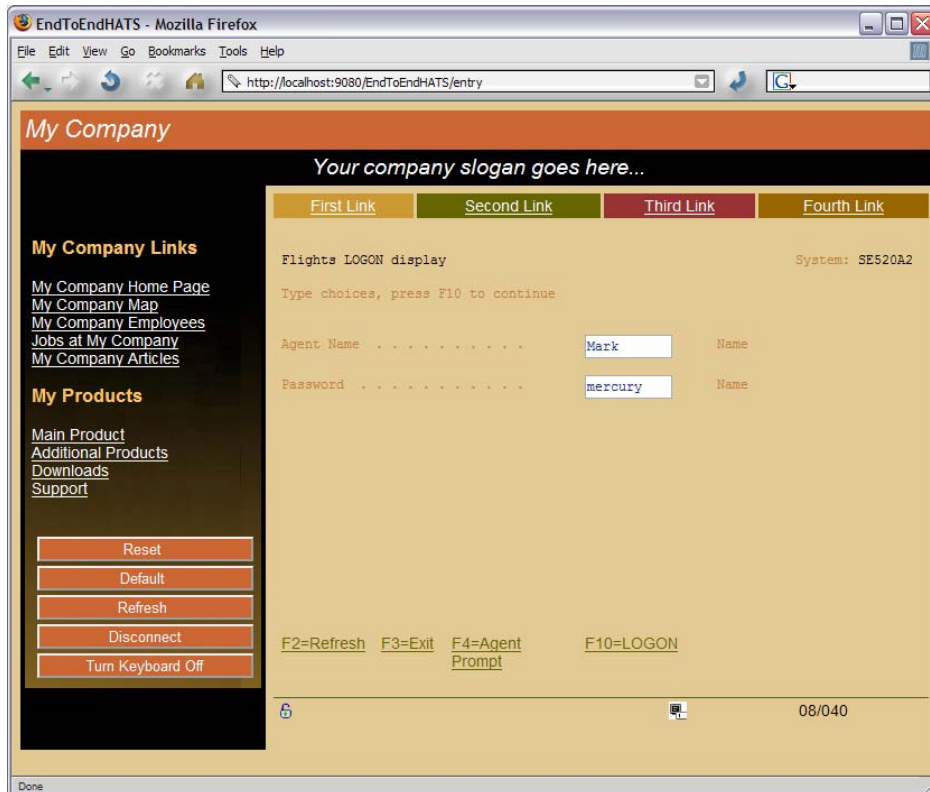


Figure 6: Agent Name and Password Web browser rendering

4. Input the required data in each entry field of the **Create Order** display as shown in Figure 7.

```
HodConn:localhost:localhost:server1:HATS_EAR51#5
Flights Reservation System - Create Order      14:21:24 7/20/05      SE520A2
Type choices, press F10 to Make Reservation

FLIGHT INFORMATION                                TICKET ORDER INFORMATION

Airline:      Flight: 0000000                    Order Number.....:  PENDING
Date of Flight...: 00 00 2005                    Customer.....:
From City.:
Depart Time.....:
To City....:
Arrival Time.....:

Class of Service - First.....: -
                  Business.....: -
                  Economy.....:  X
Number of Tickets.....: 01
Price $.....:
Tax $.....:
Total Due w/ Tax $.....:

F2=Refresh  F4=FROM Cities  F5=TO Cities  F6=Flights  F7=Customers
Buffer length longer than record for member ORDERS.
MA* ho      localhost:server1:hats_ear51#5      09/023
```

Figure 7: Create Order display

To create a new reservation, enter data into all the required fields. Either manually input each field, or press the function key, which shows a window containing a list of selections. (Refer to Figures 7 and 8.)

1. In the first field, enter the **Date of Flight**. (**Note**: The entry must be later than the current date.) Enter the date in the following format: MM DD YYYY (where MM=month, DD=day, and YYYY=year).
2. To enter the **From City**, which is the city for departure, type in the name or press **F4 (FROM Cities)** to show a list of available city names.
  - Type the first few letters of the city in the **Position To** field to set the cursor position or scroll through the item with the page up/page down key.
  - Type **1** for selecting a city from the list.
  - Press **Enter** to input the selection into the **From City** field.
  - Press **F3 (Exit)** to return to the initial screen without making a selection.
  - The **City Selection** window uses a different display file *FRS402DF*, which is invoked by the RPG program *FRS402* to get information from the *FRCITY* file.
3. Enter information in the **To City** field, which is the arrival city for the new reservation. Enter the name of the city directly or press **F5 (TO Cities)** to show the list of available city names.
  - Type **1** to select a city from the list. Press **Enter** to input the selection into the **To City** field.

- Press **F3 (Exit)** to return to the initial screen without making a selection for the *To City* field.
- This is also a different display file, *FRS403DF*. It calls the RPG program *FRS403* to get information from the *TOCITY* file.

**Note:** When using the **Position To** search in the prompted window, it is important to enter the first letter in uppercase because the search is case-sensitive. This case sensitivity applies to the other prompt screens used in this *Flight Reservation System* application as well.

4. After specifying the **Departure City** and **Arrival City**, press **F6 (Flights)** to retrieve a list of available flights.
  - Type **1** to select the flight from the list.
  - Press **Enter** to reflect the selection into the initial screen.
  - Selecting the flight will make entries for *Airline*, *Flight*, *Depart Time*, *Arrival Time*, and the *Price* fields.
5. After entering the flight information, the cursor will move to the right panel. Input the customer name manually or press **F7 (Customers)** to show the list of available customer names.
  - Type **1** to select a customer name from the list.
  - Press **Enter** to enter the selection into the **Customer** field.
  - Press **F3 (Exit)** to return to the initial screen without making entries for the **Customer** field.
  - The **Customer Selection Window** uses another display file, *FRS405DF*, which calls the RPG program *FRS405* to retrieve customer information from the *CUSTOMER* file.
6. Select the **Class of Service (First, Business, or Economy)** and enter the **Number of Tickets**. After completing all the fields, the display will look similar to Figure 8.

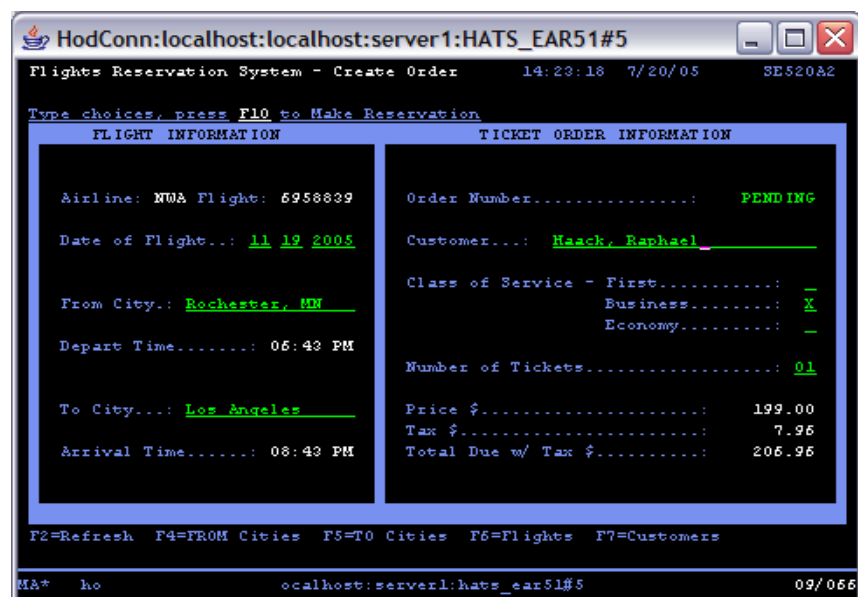


Figure 8: Create Order display

7. Submit the new reservation.

8. Check the entries (Figure 9) and press **F10** to make the reservation.

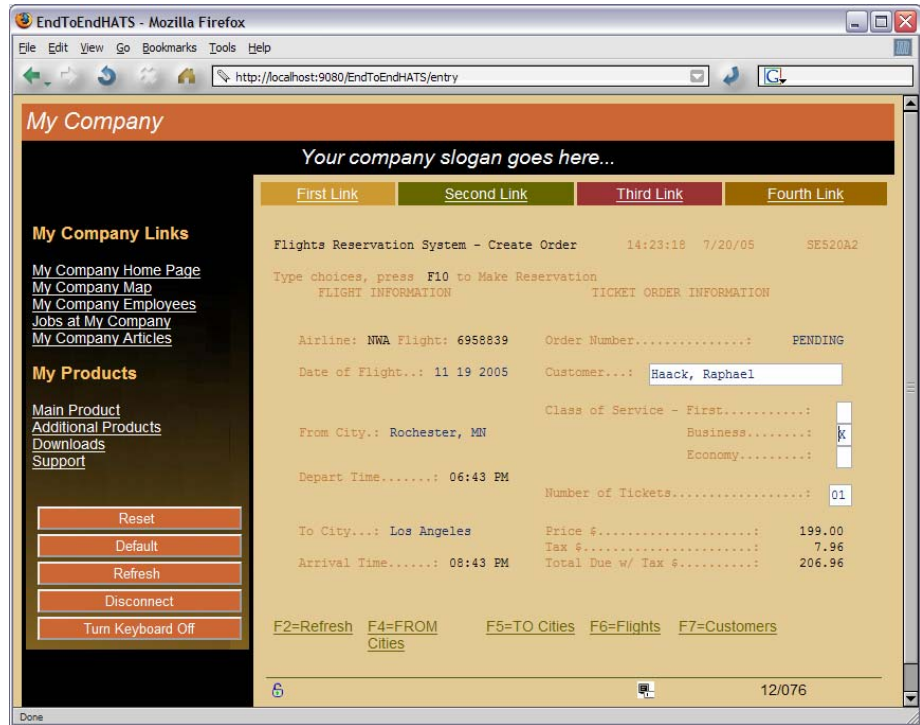


Figure 9: Create Order information verification

9. After the completion of a new reservation, a **Ticket Confirmation Window** is shown (see figures 10 and 11).

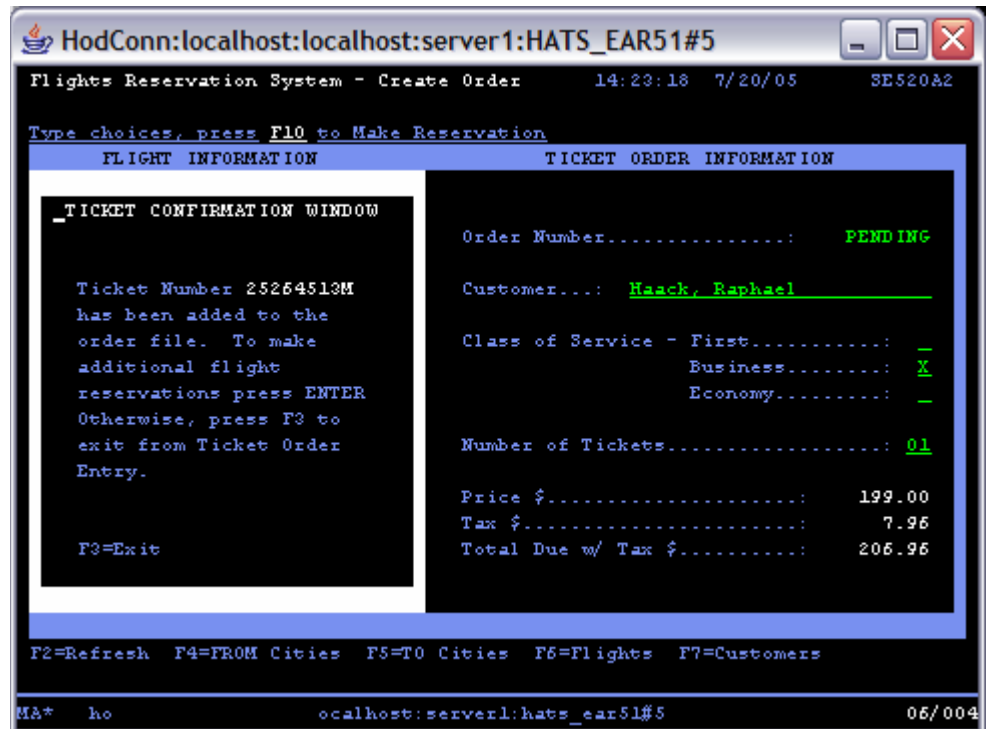


Figure 10: Green-screen Ticket Confirmation Window

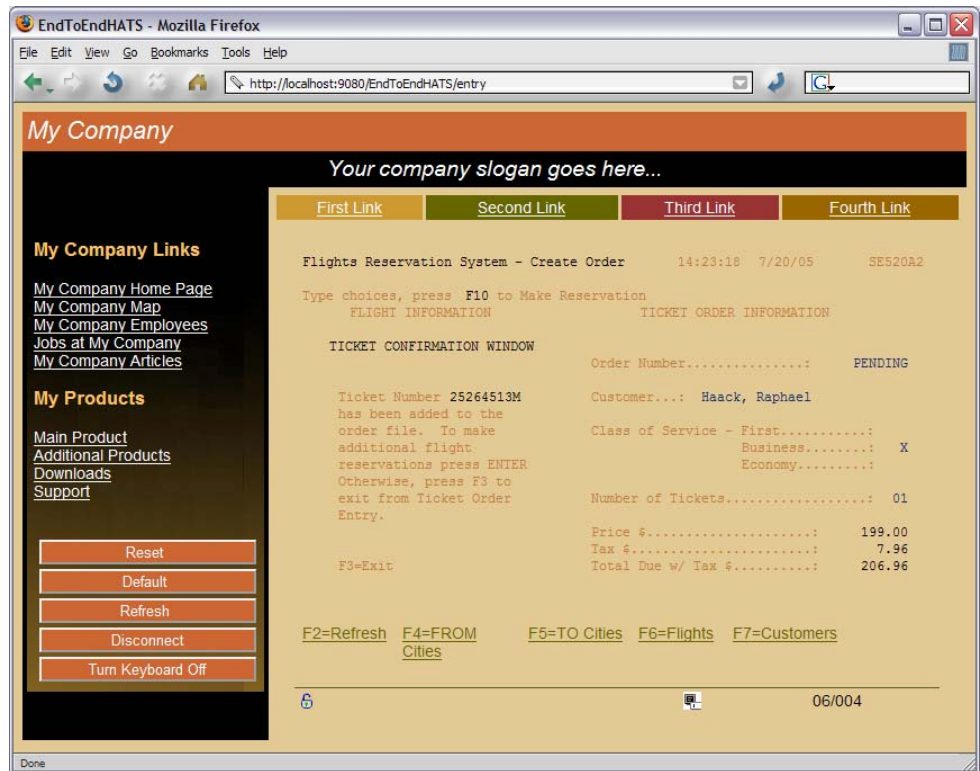


Figure 11: GUI Ticket Confirmation Window

10. Press **Enter** to create another reservation or press **F3 (Exit)** to return to the main menu.

Here, the **Ticket Confirmation Window** uses the different display file, *FRS406DF*, which calls the RPG program *FRS406* to handle the request.

**Note:** If flights are not available when pressing **F6 (Flights)**, input different cities for the **From City** or **To City** fields. Also, if the **Departure City** or **Arrival City** field is missing, it will not be possible to prompt for the **Flight Selection** window.

## Customizing the HATS project

After HATS creates a default rendering of the GUI interfaces, the interfaces can be customized to meet more specific needs. The following instructions demonstrate how to customize HATS interfaces for a specific business application.

### Customizing FLGHT400

There are a number of different ways to use HATS to enhance the FLGHT400 application. Here are just a few examples of what can be done:

#### Modifying default renderings

As seen in the green-screen and browser images below (Figure 12 and Figure 13), HATS default settings do not render all FLGHT400 settings correctly. This can easily be solved by modifying the default rendering settings of the HATS project.

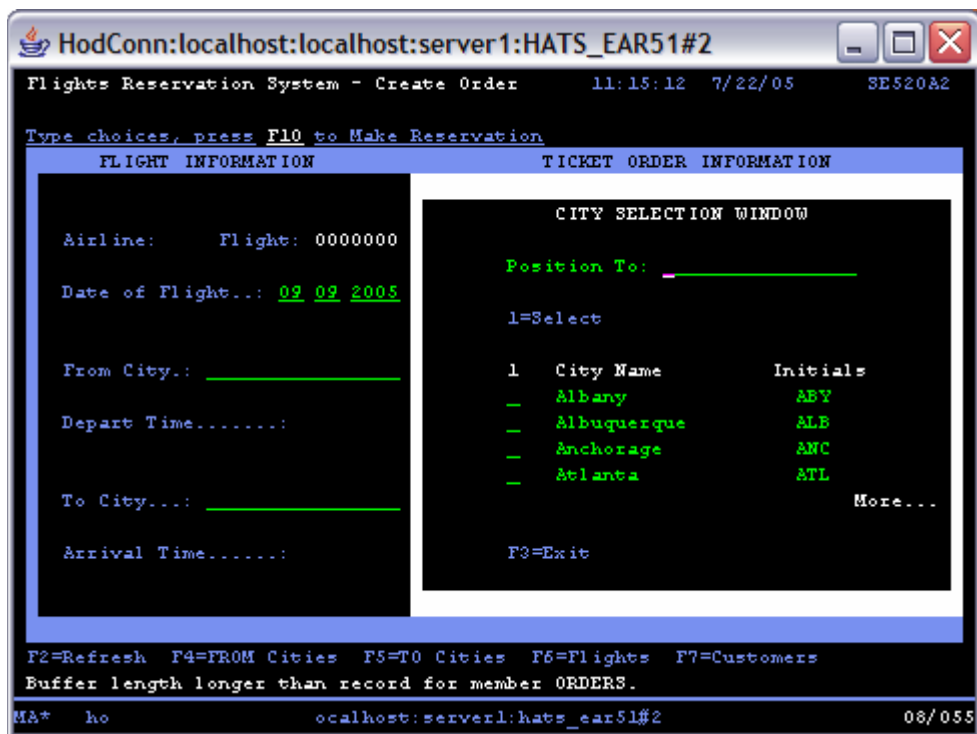


Figure 12: Green-screen images incorrectly displayed



As seen in Figure 13, the default HATS rendering does not correctly display the various subfile windows.

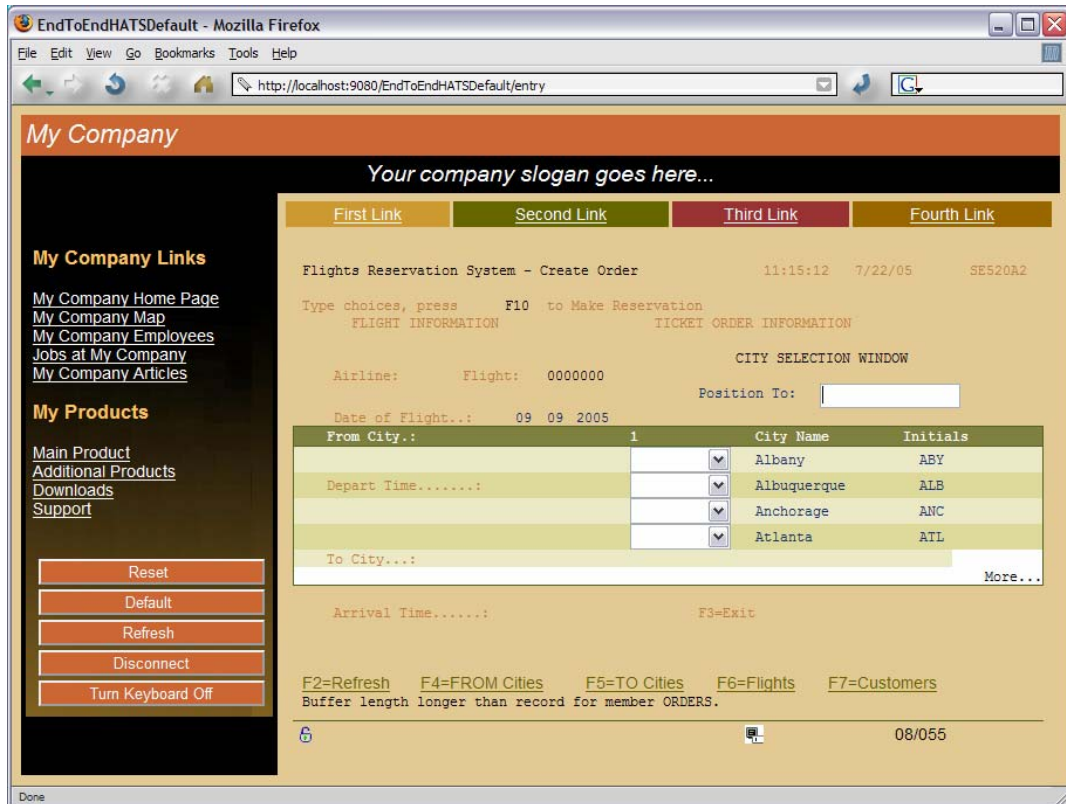


Figure 13: Incorrectly displayed subfile windows

To modify the default rendering settings to correctly display FLGHT400 information, follow these steps:

1. Expand the HATS project and double-click **Project Settings**.
2. In the **Settings** window that opens, click the **Rendering** tab (Figure 14). This is a list of the default project renderings.

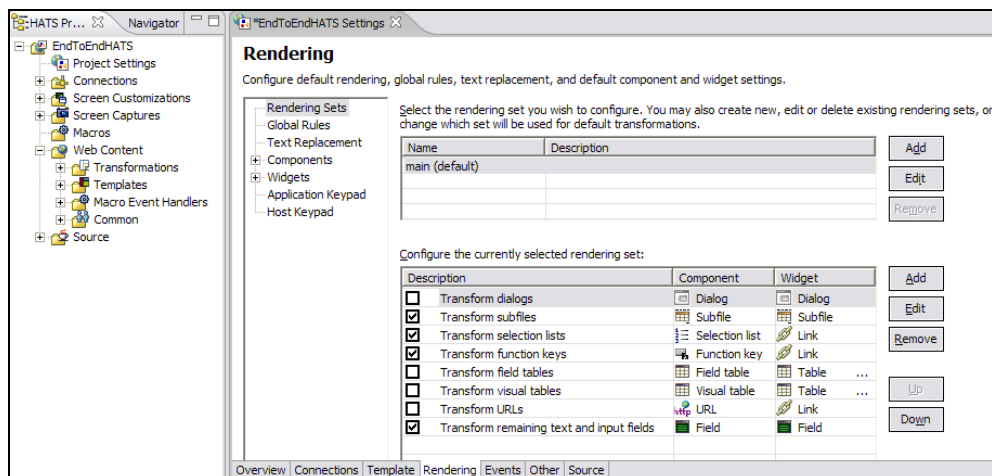


Figure 14: Rendering tab



3. Check the **Transform dialogs** option to enable the rendering.
4. Select **Transform remaining text and input fields** (as shown in Figure 15) and click **Edit**.

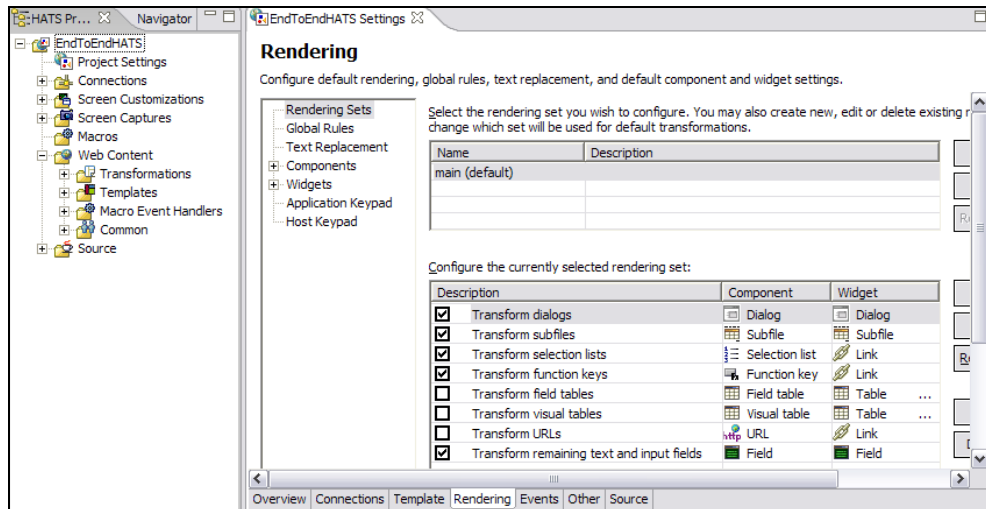




Figure 15: Transform dialogs

5. A **No Screen Captures** pop-up window will display saying that a screen capture is needed to add or edit a host component (as shown in Figure 16).
6. Click **Yes**.



Figure 16: Capture a screen

7. When the **HATS Host Terminal** window opens, click the  icon, accept the default settings, and click **Finish**.
8. Click the  icon and then close the window.
9. Select **Transform remaining text and input fields** and click **Edit**.

10. The **Edit a Default Rendering Item** window (as shown in Figure 17) will display. Click **Next**.

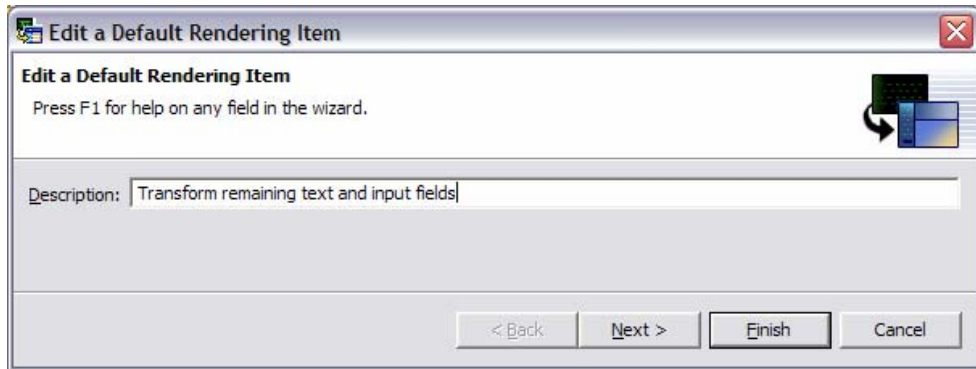


Figure 17: Edit a Default Rendering Item window

11. Click **Next** again. This brings up the **Rendering Options** window (as shown in Figure 18).

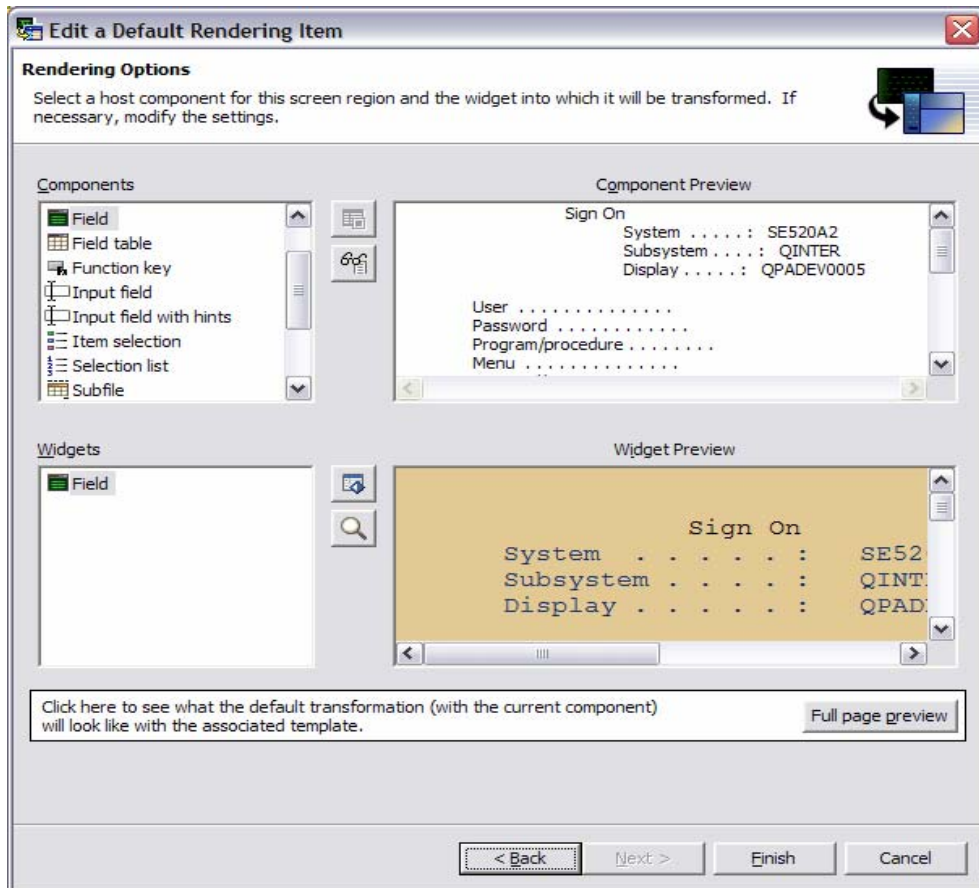


Figure 18: Rendering Options window

12. Click on the **Widget Settings** icon ()
13. In the **Settings** window, check the **Enable extended attributes** field widget setting and also clear out the **Reverse video style** field.
14. Click **Ok**.

15. Click **Finish**. With these simple changes HATS will now correctly render all of the screens in the FLGHT400 application.
16. Success (is shown in figures 19 and 20).

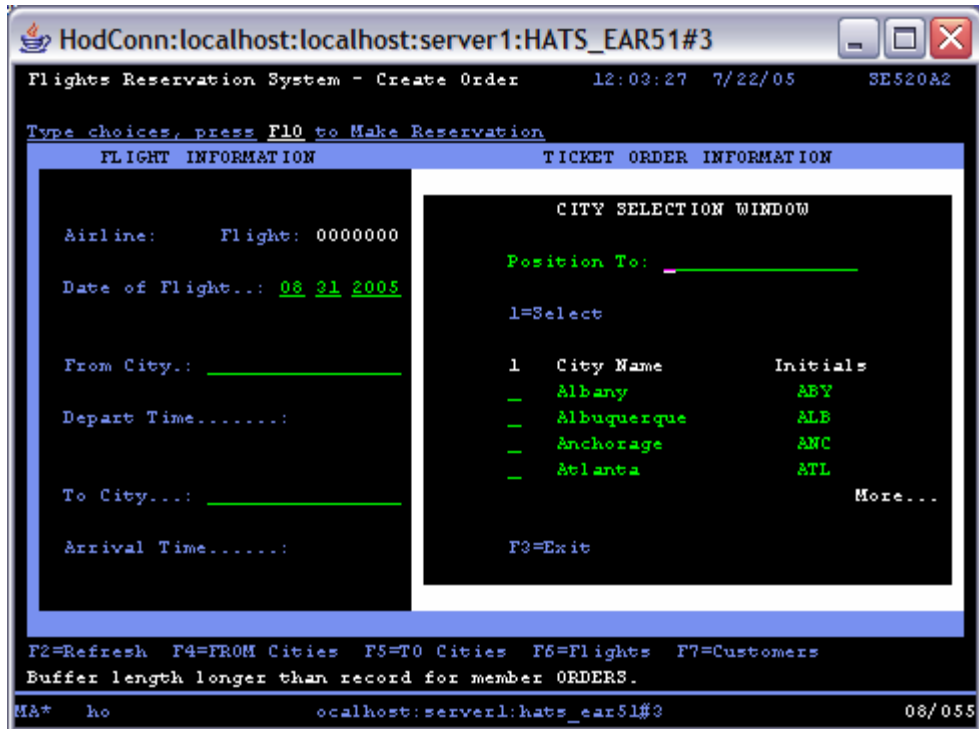


Figure 19: Green-screen successful rendering

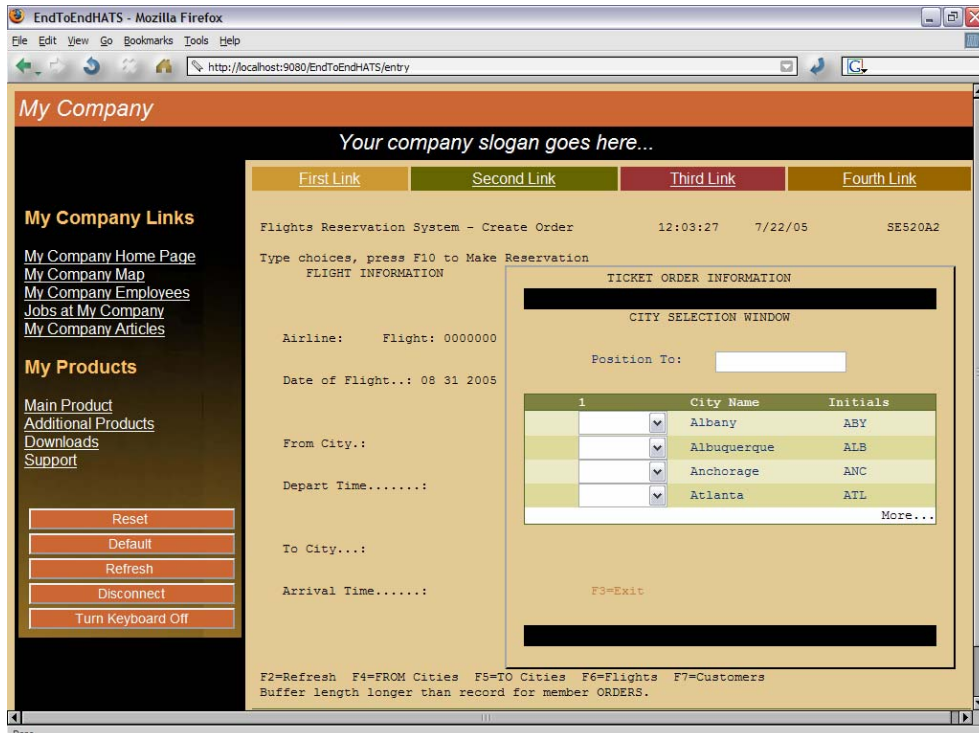



Figure 20: GUI successful rendering

### Adding a calendar pop-up

On the **Flights Reservation System – Create Order** screen of the FLGHT400 application, a date must be added across three input fields. This is easily done with the following functions that are built into HATS:

1. Create a new screen capture of the screen to be modified.
2. Click on the **Open Host Terminal** icon (  ). Log in to the iSeries system and navigate to the **Flights Reservation System – Create Order** screen (Figure 21).

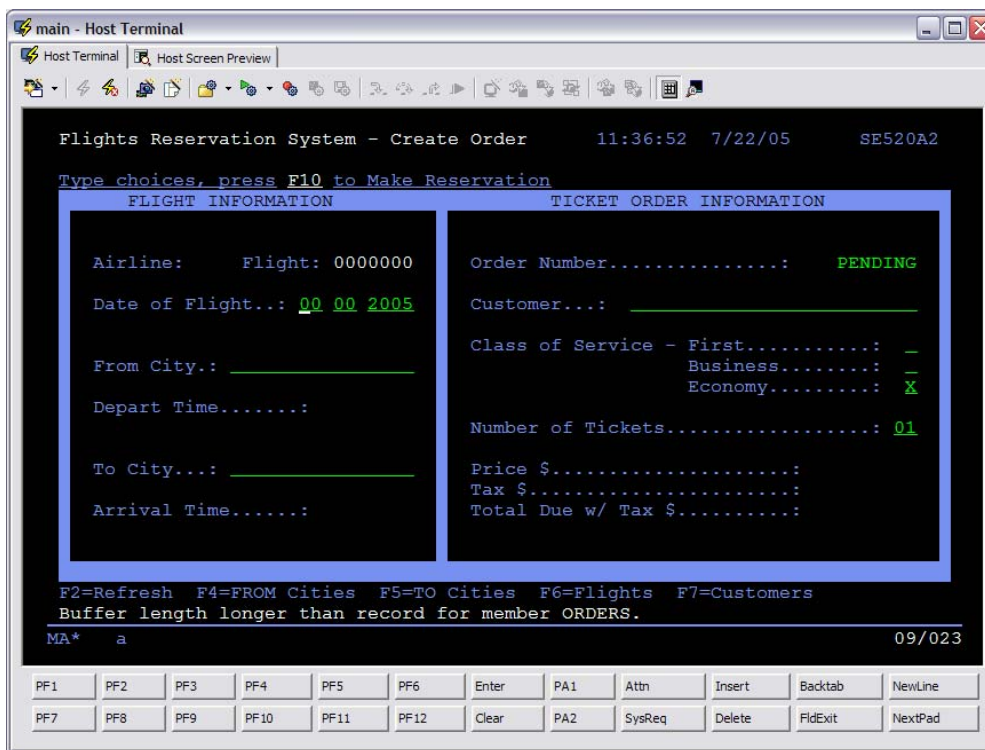





Figure 21: Flights Reservation System — Create Order screen

3. Click on the **Create Screen Capture** icon (  ).
4. Enter **FlightsReservationSystemCreateOrder** as the name and click **Finish**.
5. Click on the  icon and close the window. The screen that must be modified is now captured.
6. Click the **Create HATS Transformation** icon (  ).
7. Enter **CalendarTransformation** for the name.
8. Click **Finish**. The **Create a Screen Customization** window will open.
9. On the **Create a Screen Customization** window, click **Next**.

10. On the **Select Screen Recognition Criteria** window (shown in Figure 22), drag the mouse across the screen image to select **Date of Flight...: 00 00 2005** **Customer...**: (highlighted in yellow box).

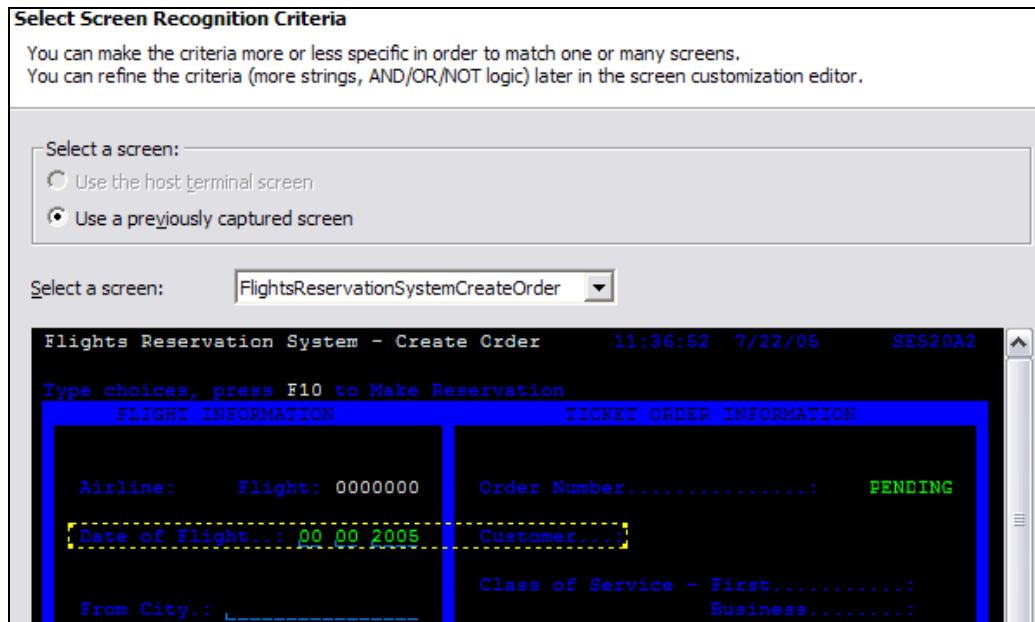


Figure 22: Select Screen Recognition Criteria window

11. Click **Finish**.
12. **CalendarTransformation.jsp** (Figure 23) will now display in the HATS project.

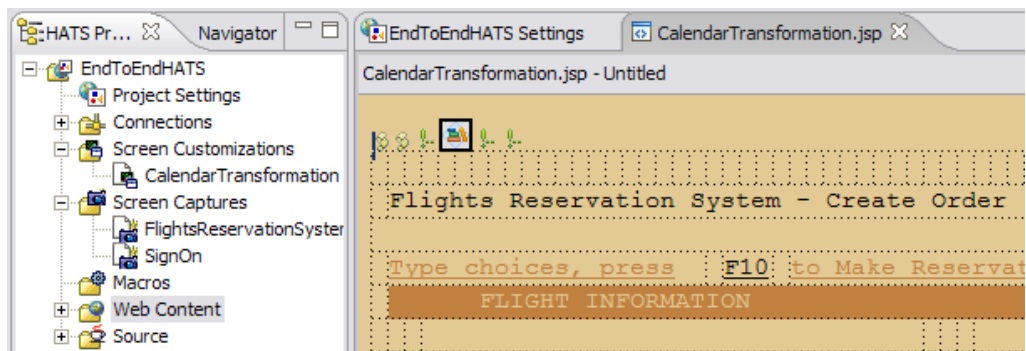
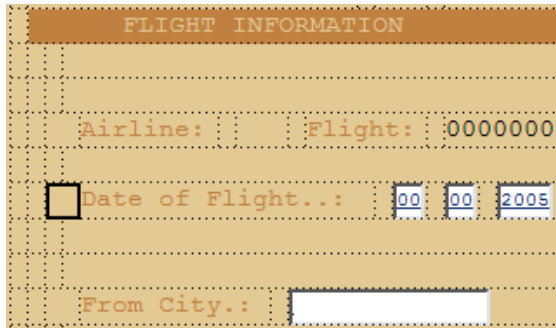


Figure 23: CalendarTransformation.jsp window

- Place the cursor in the highlighted **Date of Flight** field as shown in Figure 24.



The screenshot shows a terminal window titled "FLIGHT INFORMATION". The fields are as follows:

Airline:	Flight:	0000000
<input type="checkbox"/> Date of Flight..:	00	00 2005
From City.:		

Figure 24: Date of Flight field

- From the top menu, select **HATS Tools > Insert Host Components**. This will open the **Insert Host Component** window.
- Select the region of the host screen (**Date of Flight**) as shown outlined in yellow in Figure 25. Click **Next**.

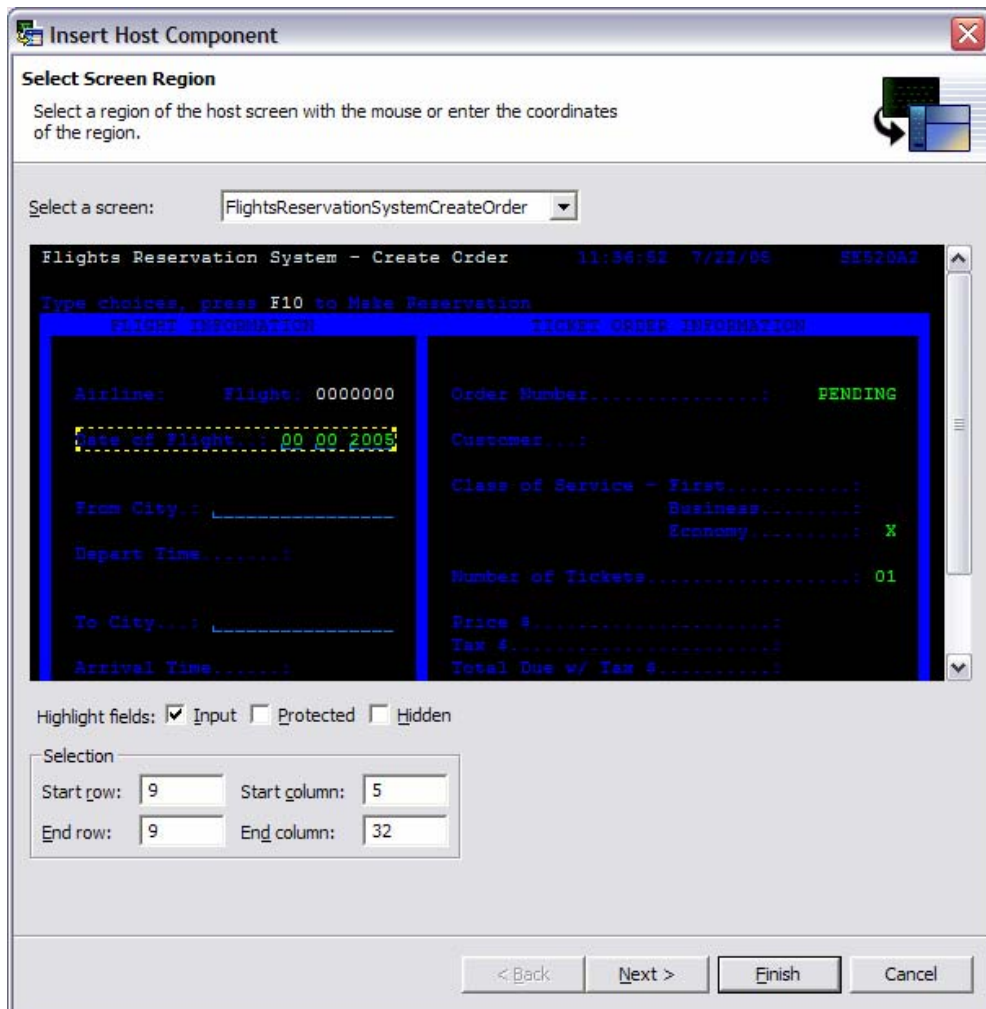


Figure 25: Date of Flight selected



16. On the **Insert Host Component Rendering Options** window (shown in Figure 26), select **Input field** under **Components** and select **Calendar** under **Widgets**.

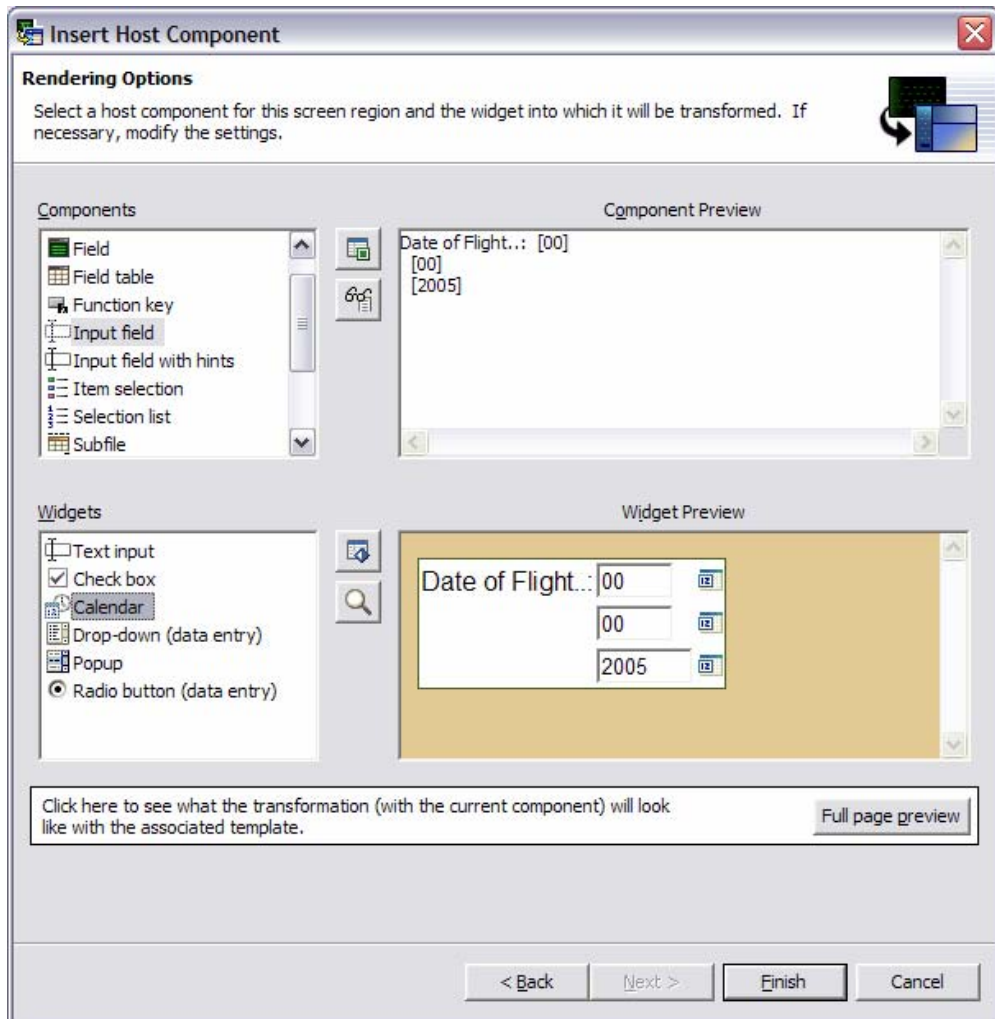


Figure 26: Insert Host Component Rendering Options window

17. Click the **Component Settings** icon () and modify the settings as shown in Figure 27.

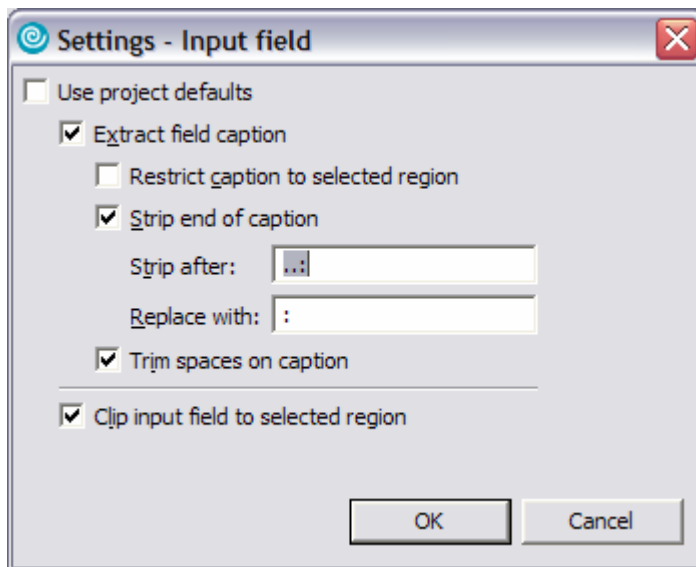



Figure 27: Component settings modified

18. Click the **Widget Settings** icon () and modify the calendar settings as shown in Figure 28. Click **OK**.

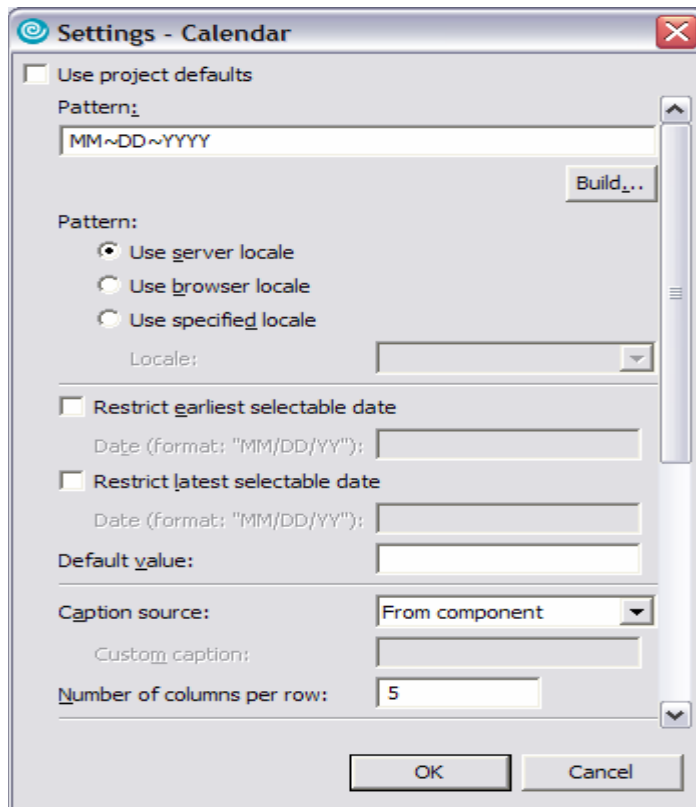
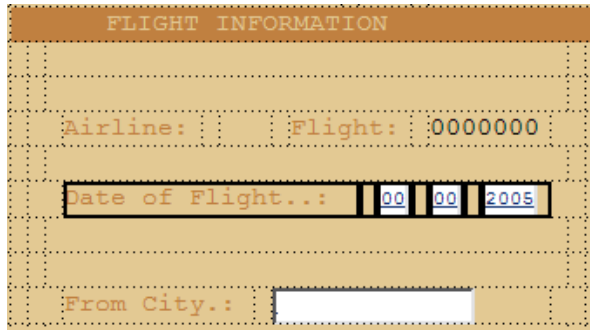


Figure 28: Calendar settings modified



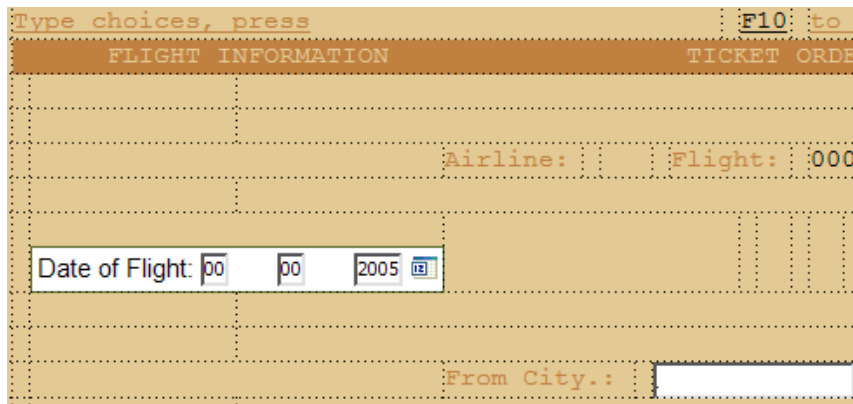
19. Click **Finish**.
20. Delete the old **Design Date of Flight..:** and the three input fields as shown in Figure 29.



The screenshot shows a terminal window with a flight information form. The form has a title bar 'FLIGHT INFORMATION'. Below it are several rows of input fields. The first row contains 'Airline:' and 'Flight: 0000000'. The second row contains 'Date of Flight..:' followed by three input boxes containing '00', '00', and '2005'. The third row contains 'From City.:'. The input fields in the second row are highlighted with a black border, indicating they are to be deleted.

*Figure 29: Input fields deleted*

21. **CalendarTransformation.jsp** will now update the window with the new calendar design as shown in Figure 30.



The screenshot shows the same terminal window as Figure 29, but with a new calendar design for the date field. The title bar now includes 'Type choices, press' and 'F10: to l'. The form has two title bars: 'FLIGHT INFORMATION' and 'TICKET ORDER'. The 'Date of Flight:' field now has a calendar icon and a small '12' icon. The input boxes contain '00', '00', and '2005'. The 'From City.: ' field is also present.

*Figure 30: New calendar design*

22. Success is shown in figures 31 and 32.

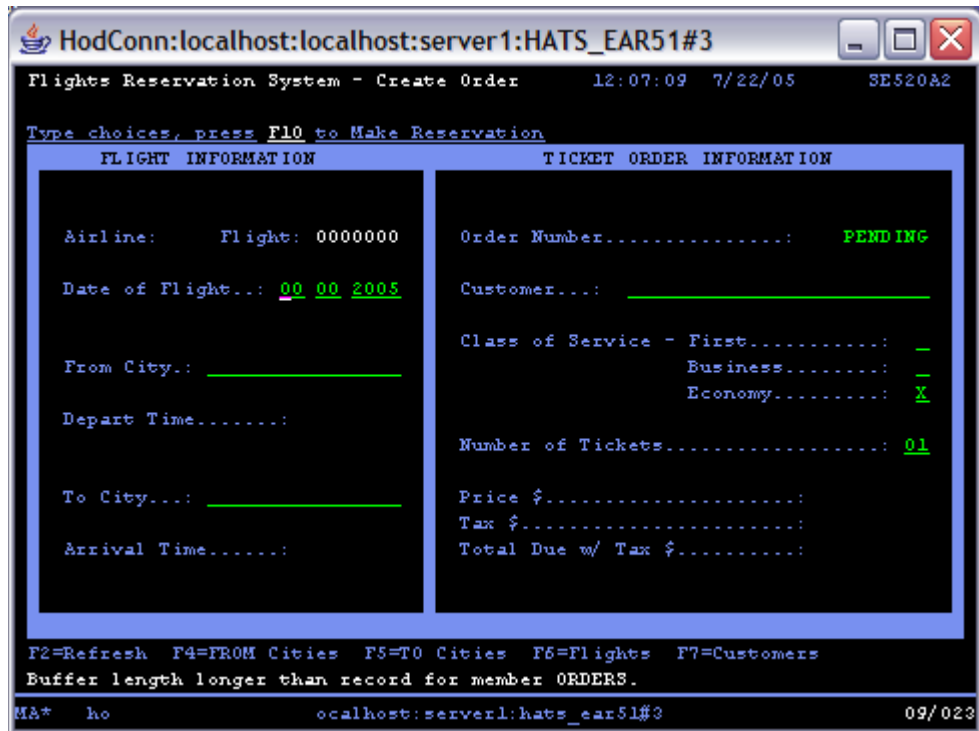


Figure 31: Green-screen success

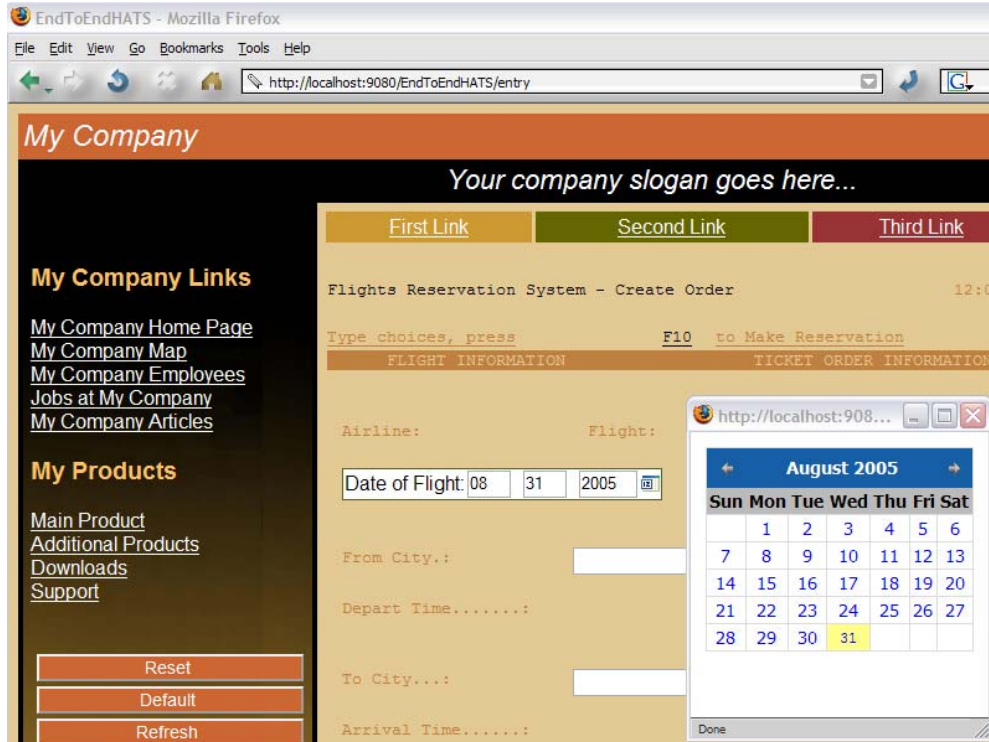


Figure 32: GUI success

## Summary

This lab shows simple modifications of the FLGHT400 application using Host Access Transformation Services (HATS). It also demonstrates the broad range of enhancements that are available by using HATS. By utilizing HATS, a green-screen terminal application can be automatically transformed into a Web-based, thin-client application that can be accessed through a Web browser on the user's personal computer.

By default, HATS also renders iSeries components as Web widgets. This includes function keys, links, buttons, and menus, each of which becomes a radio button list, link list, or drop-down list option.

These features represent a small portion of the available HATS functions. Documents and other resources are available to learn more about this transformational tool through links provided in the "Additional information" section of this paper.

## Additional information

These Web sites provide useful reference materials to supplement the information contained within this document:

- Host Access Transformation Services (HATS) Roadmap  
[www.developer.ibm.com/vic/hardware/myportal/develop/roadmap](http://www.developer.ibm.com/vic/hardware/myportal/develop/roadmap)
- WebSphere Host Access Transformation Services product page  
[ibm.com/software/webservers/hats/index.html](http://ibm.com/software/webservers/hats/index.html)
- IBM WebSphere Host Access Transformation Services (HATS) demonstrations  
[http://websphere.dfw.ibm.com/atdemo/atdemo\\_hats.html](http://websphere.dfw.ibm.com/atdemo/atdemo_hats.html)
- Host Access Transformation Services (HATS) V6 development lab  
[ibm.com/servers/enable/site/education/labs/4202/4202.pdf](http://ibm.com/servers/enable/site/education/labs/4202/4202.pdf)
- IBM WebSphere Host Access Transformation Services (HATS) V6 Information Center  
<http://publib.boulder.ibm.com/infocenter/hatsv6/index.jsp>
- iSeries Application Innovation Program  
[ibm.com/servers/enable/application/innovation](http://ibm.com/servers/enable/application/innovation)
- FLGHT400 overview  
[ibm.com/servers/enable/site/ideveloper\\_j2ee/etoe/pdfs/flight400\\_overview.pdf](http://ibm.com/servers/enable/site/ideveloper_j2ee/etoe/pdfs/flight400_overview.pdf)
- IBM Software Access Catalog  
[www.developer.ibm.com/isv/welcome/softmall.html](http://www.developer.ibm.com/isv/welcome/softmall.html)
- Trial version of Rational Web Developer for WebSphere Software V6.0  
[ibm.com/developerworks/downloads/r/rwd/?S\\_TACT=105AGX14&S\\_CMP=DWNL](http://ibm.com/developerworks/downloads/r/rwd/?S_TACT=105AGX14&S_CMP=DWNL)
- Trial version of IBM WebSphere Host Access Transformation Services Toolkit  
[http://www14.software.ibm.com/webapp/download/preconfig.jsp?id=2004-08-19+09%3A27%3A47.465987R&S\\_TACT=104CBW71&S\\_CMP=&s=](http://www14.software.ibm.com/webapp/download/preconfig.jsp?id=2004-08-19+09%3A27%3A47.465987R&S_TACT=104CBW71&S_CMP=&s=)

## About the author

**Michael Sandberg** is a technical consultant in the IBM eServer Solutions Enablement team, located in Rochester, Minnesota. For the past four years, he has been involved in supporting iSeries solution providers as they modernize and innovate their applications. As part of this work, he has accumulated technical expertise in WebSphere Application Server, WebSphere Portal Server, WebSphere Development Studio, and other technologies that focus on iSeries application innovation.

## **Trademarks and special notices**

© IBM Corporation 1994-2005. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, the IBM logo, eServer, iSeries, WebSphere, Rational, AS/400, and ibm.com.

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, other countries, or both.

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

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.