CHAPTER 2

Basic SESM Customization and Development

In some Cisco SESM deployments, you can use one of the sample SESM web applications as a starting point and accomplish look-and-feel modifications to the JSP pages without changing the preprogrammed SESM software. This chapter explains how to do basic customization of the look-and-feel elements of an SESM web application. The chapter discusses these topics:

The SESM software uses Java resource bundles and properties files for localization of text, labels on web-page controls, and messages. For information on localization and resource bundles, see Chapter 5, "SESM Internationalization and Localization."

Customizing an SESM Web-Application: An Overview

When you develop a Cisco SESM web application, three levels of web-application customization are possible: basic, advanced, and system integrator.

Basic SESM Customization

In some Cisco SESM deployments, you can use one of the sample SESM web applications as a starting point and accomplish look-and-feel modifications to the JSP pages without changing the preprogrammed SESM software. In this type of customization, the developer might change text, images, and the positioning of elements on a page. Basic customization might be appropriate when using SESM for Small-to-Medium-Size Enterprise (SME) wireless LAN hotspots that require simple look-and-feel changes to the web application or require some subset of SESM functionality, such as authentication.

This level of customization is also used by deployments whose business requirements correspond closely to the existing structure of a sample SESM web application, such as NWSP. This chapter discusses basic customization.

Advanced SESM Customization

Other SESM deployments will require use of the more advanced customization techniques. In this type of SESM web-application customization, you might add or remove JSP pages, or move elements from one JSP page to another. You might add a JSP-page dimension decorator for the SESM user-shape mechanism. You can accomplish some of the more advanced customizations by modifying the deployment descriptor file (web.xml) for the SESM web application. You accomplish other advanced customizations by creating new JSP pages. No Java servlet programming is needed. For information on advanced customization techniques, see Chapter 3, “Advanced SESM Customization.”

System-Integrator Customization

Some service-provider deployments of SESM may need the professional services of system integrators to modify the functionality of the internal SESM software, such as the Java servlets for the SESM controls. This type of customization is beyond the scope of this guide and requires the involvement of Cisco technical assistance.

Changing the Look-and-Feel Elements

Basic SESM web-application customizations involve changing the look-and-feel elements to meet the service provider’s brand requirements. The customizable JSP pages and look-and-feel elements allow developers to create web pages that incorporate artistic flexibility and meet corporate identity standards without requiring extensive JSP or Java programming expertise.

To meet the service provider’s corporate standards, you can modify static markup language elements and images (template text) that appear in a template or JSP page. For example, you can change the static HTML elements for text and formatting. In these elements, the developer can:

- Change the background colors and background images used in the JSP pages
- Adjust the layout of the JSP pages
- Modify or replace icons, images, and graphical elements used in the JSP pages
- Customize a style sheet

If you are performing only basic look-and-feel customizations, you must avoid changing programmatic elements within the JSP pages. Do not change directives and code in scripting elements such as JSP expressions, scriptlets, and declarations.

Customizing some of the SESM functionality that is preprogrammed into the JSP pages can be accomplished by configuring the application. Other functionality changes may require modifying the code in the JSP pages. For information on these types of customization, see Chapter 3, “Advanced SESM Customization.”

Using a Sample SESM Web Application

Each sample Cisco SESM web application, such as New World Service Provider (NWSP), includes a fully functional set of web components. Each sample web application uses the same SESM infrastructure. The simplest, fastest-to-implement approach to developing an SESM web application is to use the components in a sample SESM web application and then change the look-and-feel elements. This section provides an overview of the set of components in the sample SESM web applications and focuses on the NWSP web application.

The sample NWSP, PDA, and WAP web applications share many of the same components.

The sets of infrastructure components provided in the PDA and WAP sample web applications are identical to the NWSP set of infrastructure components. However, because the PDA and WAP web applications are designed for specific client devices, they use JSP pages that are different from the JSP pages of NWSP.

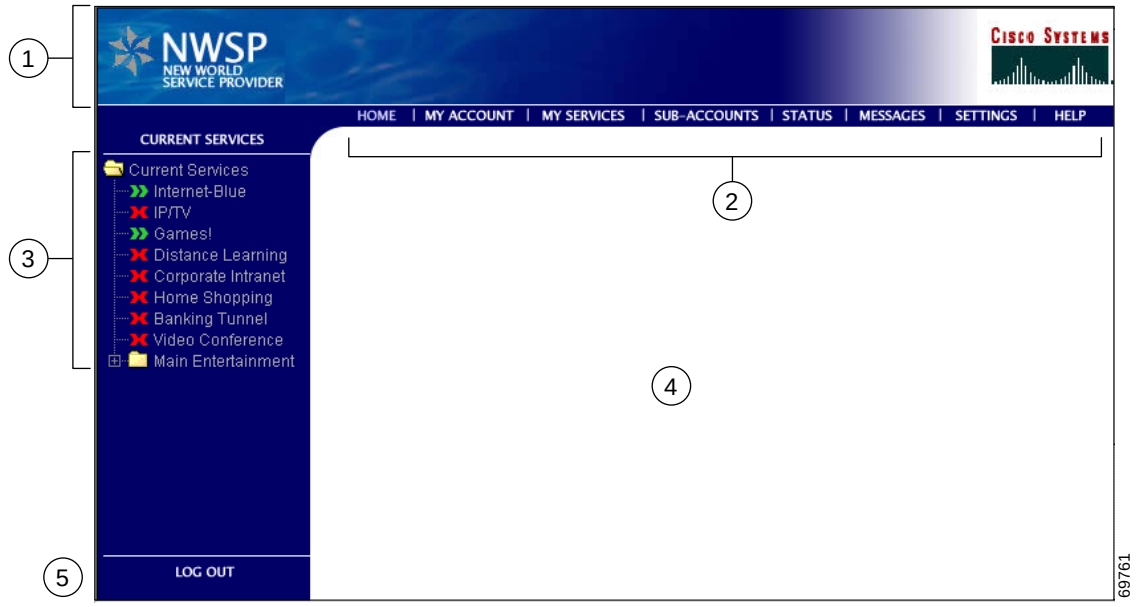
The sets of resources (Java resource bundles) used in the PDA and WAP sample web applications are very similar to the NWSP set of resources. The deployment descriptor files used for the NWSP, PDA, and WAP applications are also very similar.

For more detailed information on the elements that appear on the JSP pages associated with a sample SESM web application, see Chapter 4, “Sample SESM Web Applications.”

NWSP User Interface

The NWSP user interface provides a web portal for network services. The subscriber uses the web portal for subscribing to and selecting services, changing account details, creating subaccounts, and viewing session status and messages. Figure 2-1 shows the home page from the NWSP web application’s user interface.

Figure 2-1 NWSP Home Page



1	Banner
2	Navigation bar
3	Service list
4	Body
5	Log Out button

The Home page and many other pages in the NWSP web application are organized into five parts:

- **Banner**—One or more images for product and brand identification.
- **Navigation bar**—A set of buttons that link to other pages where the subscriber can perform tasks such as service subscription or view information such as service status.
- **Service list**—A tree of icons and associated service names that the subscriber uses to select subscribed services from a tree of services and service groups.
- **Body**—An area for the content provided by the NWSP web application. Forms for subscriber tasks such as account management, service subscription, and subaccount creation appear in the body area. (With NWSP, network services are displayed in a new browser window.)
- **Log Out button**—A button to log out of the NWSP web application.

JavaServer Pages

The sample SESM web applications are implemented in a set of JSP pages and Java servlets. To change the look and feel of an SESM sample web application, you need to become familiar with the elements that are present in the JSP pages provided in a sample SESM web application such as NWSP.

The JSP pages in the NWSP web application include a complete set of customizable images, background colors, icons, buttons, a navigation bar, and style sheets. For NWSP, the Fireworks graphics design tools were used to create the icons for the service list and the buttons for the navigation bar.

Banner Images and Background Colors

In the NWSP banner (see Figure 2-1), the images and background colors are used for branding. You can replace these images with company, product, and brand logos that are appropriate for your deployment.

Icons and Buttons

For the NWSP service list and navigation bar (see Figure 2-1), the icons and buttons are provided in various formats, including GIF, JPEG, and PNG.

- The GIF and JPEG image files are incorporated into the web pages. Their small file size makes the optimized GIF and JPEG files suitable for downloading. You can also use these files to edit the images. You can use Fireworks or another graphics design tool, such as Photoshop, to customize the images and embedded text.
- The PNG files are used when the web designer edits the images. Because PNG format is the native Fireworks file format, you can use Fireworks to customize the images and embedded text. PNG files can also be read by other graphics applications, such as Photoshop.

Buttons, such as the Log Out button and Cancel button, that do not appear in the NWSP navigation bar are defined once in a single JSP page or in a Dreamweaver template. You can customize each button in the one location where it is defined and have the change take effect in all JSP pages that use the button.

Navigation Bar

A Dreamweaver navigation bar (sometimes called a nav bar) is a set of buttons that appears on a series of related web pages and that provides a consistent mechanism for navigation between pages. For example, the NWSP web application contains a navigation bar (see Figure 2-1) below the banner on most of its pages.

When modifying a Dreamweaver navigation bar, you can use the existing NWSP buttons or create a new set of buttons. You can use Fireworks or another graphics design tool to change the text on the existing NWSP buttons. For more information on the navigation bar, see the "NWSP Navigation Bar" section on page 4-15.

Style Sheets

The NWSP web application uses Cascading Style Sheets for its presentation elements. The style sheets define classes for the textual elements that are used on the JSP pages. The style sheets allow designers to change fonts, margins, colors, and other aspects of style that control the layout and design of the NWSP web pages. The style sheets used by NWSP are located in the *install_dir*\nwsp\docroot\styles directory. In other SESM web applications such as PDA, there may be multiple brand-specific style sheets in different directories from which the SESM software dynamically selects one style sheet based on the subscriber's brand.

Dreamweaver Templates

The sample NWSP web application includes two Dreamweaver templates. These files have the suffix `.dwt` and reside in the /nwsp/docroot/templates directory. Dreamweaver templates can be very useful for customizing or maintaining a web application's JSP pages when many pages have the same layout. By modifying a template and then updating the JSP pages that use the template, you can change the look and feel of an entire set of pages very quickly.

When a JSP page is derived from a template, the JSP page has locked regions that cannot be edited and editable regions that can be edited. The intention with locked regions is that if changes are required on a locked region, the changes are made in the template file. After the changes are made to the template, all files that use the template can then be automatically updated to incorporate the changes. You modify the template and update all occurrences on the web site using the Dreamweaver **update** commands in the Modify > Templates menu.

The use of Dreamweaver templates and automatic updating is a recommended approach but not a requirement.

If changes are required to individual JSP pages (as opposed to in a Dreamweaver template), the part of the individual JSP page that you can modify appears between BeginEditable and EndEditable comments. For example, the main editable area in the mainTemplate.dwt is empty, allowing each JSP page to provide content that is appropriate for its purpose:

```
!-- #BeginEditable "main" -->
 
<!-- #EndEditable -->
```

For more information on templates, see the "NWSP Templates" section on page 4-10 and the Dreamweaver UltraDev documentation.

Developing an SESM Web Application

Depending on the service provider's business requirements for an SESM web application, the steps required to develop the application may vary. These steps are described in the following sections:

Tip One error that developers frequently commit when developing an SESM web application is that they do not perform the steps required *to compile* the JSP pages after they have modified them. The SESM sample web applications are installed with *precompiled JSP pages*. After changing the JSP pages, you must take a few simple steps to recompile them successfully. For information on compiling JSP pages, see the "JSP Compilation" section on page 2-8.

Defining the Business Requirements

The process of defining the business requirements for a specific service provider's SESM web application can be organized around the following questions:

- What functionality is required?
- What look-and-feel elements (icons, images, background colors, and style sheets) must be customized?
- Will SESM web pages be rendered to match one or more subscriber characteristics, such as device or brand?

- What types of localization (if any) are required?
  - If only English-language subscribers will use an SESM web application, the base set of components in a sample SESM web application may not require localization.
  - If subscribers use a single language other than English, a single web site localized for this language may be sufficient.
  - If subscribers use many languages, rendering SESM web pages localized for each subscriber's language and character set may be required.
- Do any application-specific messages need to be internationalized and localized?

Designing and Implementing an SESM Web Application

The design and implementation phases may involve one or more of the following tasks:

- Modifying the functionality of a sample SESM web application
- Customizing the look and feel of web elements such as icons, images, background colors, and style sheets
- Localizing web elements
- Creating additional locale-specific properties files
- Designing, implementing, and populating a sparse-tree directory
- Coding revised or new JSP-pages dimension decorators for the user-shape mechanism
- Configuring the web application deployment descriptor file
- Internationalizing and localizing exceptions

SESM Class Libraries

The Cisco SESM software provides several specialized Java class libraries that encapsulate the functionality required in an SESM web application. The SESM class libraries are distributed in the JAR (Java archive) files listed in Table 2-1. In the table, *install_dir* is the directory where the SESM software is installed, and *webapp* is a directory where a sample SESM web application, such as NWSP, is installed.

Table 2-1 JAR Files for an SESM Web Application

JAR File	Description
install_dir/lib/lib/ com.cisco.sesm.lib.jar	Classes for Java Management Extensions (JMX), JUnit utilities, and logging facilities used by an SESM web application.
install_dir/*webapp*/docroot/Web-inf/lib/ com.cisco.sesm.contextlib.jar	Classes for the SESM decorators and controllers, internationalization and localization, and tag libraries.
install_dir/*webapp*/docroot/Web-inf/lib/ jsp.jar	Classes for the precompiled JSP pages.
install_dir/*webapp*/docroot/Web-inf/lib/ sesm.jar	Classes for core and model services such as authentication, authorization, service connection, and billing.

The CLASSPATH environment variable must be set to *install_dir*/*webapp*/docroot/Web-inf/lib to tell the Java compiler the location of the com.cisco.sesm.lib.jar file. In the sample SESM web applications, the environment variable is set through the start script (for example, startNWSP.sh on UNIX and startNWSP.cmd on Windows), which is executed when the web application is started. Any JAR file, including com.cisco.sesm.contextlib.jar, jsp.jar, and sesm.jar files, can be found by the Java compiler if it resides in the web application's /Web-inf/lib directory.

Javadoc Documentation

The Cisco SESM software includes a full set of online documentation in Javadoc format. The Javadoc documentation includes information on all Java classes that implement the SESM software. If you are customizing the JSP pages of an SESM web application, the Java class descriptions for control servlets, decorator servlets, JavaBeans, and tag libraries are of particular interest. You can access the Javadoc by opening the file *install_dir*\docs\apidoc\index.html in a browser.

JSP Compilation

The use of precompiled JSP pages allows an SESM server that has the required Java Runtime Environment (JRE) to run the NWSP web application. If an SESM server has the JRE, it can run an SESM web application without having a Java 2 SDK installed.

The JSP pages in each sample SESM web application are precompiled and the resulting servlet classes are installed in the *install_dir**webapp*\docroot\Web-inf\lib\jsp.jar file. In the path name, *install_dir* is the directory where the SESM software is installed, and *webapp* is a directory where a sample SESM web application, such as NWSP, is installed.

A Java 2 SDK is not required to run an SESM web application. However, to perform development on the JSP pages, you *must* install the Java 2 SDK on the development machine. The Java 2 SDK is necessary for recompiling changed JSP pages. For information on the required Java 2 SDK, see the "Hardware and Software Requirements for Development" section on page 1-6.

Tip On machines that are used for SESM web application development, we recommend that you install the Java 2 SDK before you install the SESM software. In that way, the SESM installation program uses the Java 2 SDK in the SESM web application startup scripts, rather than a JRE.

To check whether a Java 2 SDK is installed on the development machine, look for a tools.jar file in the *JDK_install_dir*/lib directory, where *JDK_install_dir* is the location where you think the Java 2 SDK is located. If the tools.jar file is present, a Java 2 SDK is installed.

If you install the Java 2 SDK *before* installing SESM, read the "Recompiling and the web.xml File" section on page 2-9.

If you need to install the Java 2 SDK *after* installing SESM, read the "Installing a Java 2 SDK After Installing SESM" section on page 2-8 and the "Recompiling and the web.xml File" section on page 2-9.

Installing a Java 2 SDK After Installing SESM

If you install the Java 2 SDK *after* installing SESM, you must do the following:

- Ensure that the JDK_HOME environment variable points to the directory where you installed the Java 2 SDK.
- Edit the SESM application startup script to use the Java 2 SDK.

The SESM application startup scripts are named start*WEBAPP*.cmd on Windows-based installations and start*WEBAPP*.sh on UNIX-based installations. In the script name, *WEBAPP* is the name of the web application. For example, the startup script for NWSP is startNWSP.cmd or startNWSP.sh.

In the startup script on a Windows-based installation, change the line that specifies the Java 2 SDK location (JDK_HOME) to point to the correct directory, and uncomment the two lines that check for a Java compiler (JAVAC). For example:

```
rem find some java or other
set JDK_HOME=D:\jdk1.3

set JAVA=%JDK_HOME%\bin\java.exe
set JAVAC=%JDK_HOME%\bin\javac.exe

rem Note that we only need this if we are developing, so for now
rem we can comment the check out. Developers should remove these comments
rem if exist "%JAVAC%" goto gotjdkhome
rem echo JDK_HOME does not point to a valid JDK. JSPs will not be compiled.
```

In startup script on a UNIX-based installation, change the line that specifies the Java 2 SDK location (JDK_HOME) to point to the correct directory, and uncomment the four lines that check for a Java compiler (JAVAC). For example:

```
# Check we can find a suitable version of the JDK
JDK_HOME=/usr/java1.2

JAVAC=$JDK_HOME/bin/$JAVACEXE
JAVA=$JDK_HOME/bin/$JAVAEXE

# Note that we only need this if we are developing, so for now
# we can comment the check out. Developers should remove these comments
if [ ! -x $JAVAC ]
then
     echo WARNING: JDK_HOME does not point to a valid JDK. JSPs can not be compiled.
fi
```

Recompiling and the web.xml File

The web.xml file in *install_dir**webapp*\docroot\Web-inf is the SESM web application deployment descriptor file. This file defines how the Jetty web server finds the JSP pages. The default web.xml file for each sample SESM web application specifies that the web server uses the precompiled JSP pages. To compile modified JSP pages, use the web.recompile.xml file in place of the default web.xml file.

Note Until you perform the following steps, changing the JSP pages in *install_dir*/*webapp*/docroot has no effect on the HTML pages that the NWSP web application displays.

To recompile modified JSP pages in the NWSP web application, you must do the following:

Step 1 Stop the web server.

Step 2 In the *install_dir*\nwsp\docroot\Web-inf directory, rename the web.xml to web.xml.bak.

Step 3 In the *install_dir*\nwsp\docroot\Web-inf directory, rename web.recompile.xml to web.xml.

Step 4 Restart the web server.

Demo Mode

You can install or configure the SESM software for demonstration mode (Demo mode) to observe how the NWSP web application works. In Demo mode, the NWSP web application can demonstrate most SESM functionality. For information on installing or configuring for Demo mode, see the *Cisco Subscriber Edge Services Manager and Subscriber Policy Engine Installation and Configuration Guide*.

Note The sample PDA and WAP demo applications also work in Demo mode. However, because the PDA and WAP applications demonstrate only the subset of SESM functionality that is appropriate for their targeted devices, this description of Demo mode focuses on the NWSP web application.

Demo mode is also useful during some phases of web-application development because this mode does not require other system components, such as a configured SSG.

- If the web designer modifies the look and feel of the NWSP web application, the changes can be viewed in Demo mode to observe the results.
- If the web developer changes programmatic pieces of a JSP page, many changes in web-application behavior can be initially tested in Demo mode to verify the results.

For the NWSP sample application, Demo mode uses a Merit RADIUS file for subscriber, service, and service group information. By default, the Merit RADIUS file is named demo.txt and resides in the *install_dir*\nwsp\config directory. A Merit RADIUS file is an ASCII file that you can modify using a text editor. Using Demo mode, you can observe how changes to a subscriber, service, or service group profiles affect the NWSP web application.

Tip If you make changes to the demo.txt file, you will not be able to observe the changes until the web server is stopped and restarted.

Captive Portal and Demo Mode

The full set of behaviors of the Captive Portal solution cannot be observed in Demo mode because a configured SSG and its TCP Redirect feature are required to make the solution work. For this reason, the Captive Portal software is not installed in a Demo mode installation of SESM.

To run Captive Portal in Demo mode, you must install the Captive Portal software in RADIUS or LDAP mode, and then start the Captive Portal web application using the **-mode** option. For example:

```
startCAPTIVEPORTAL.sh -mode Demo
```

With Captive Portal running in Demo mode, you can use simulated HTTP requests to view the behavior of the Captive Portal servlet and the content web applications. For example, you could send a simulated HTTP request to the message portal servlet (`MessagePortalServlet`) if you included the required query-string parameters (such as `CPURL` and `CPDURATION`) in the request. For information on the Captive Portal solution and the query-string parameters, see the "Captive Portal Web Application" section on page 4-23.

RADIUS Mode Functionality

To simulate RADIUS mode functionality (such as service selection), the subscriber, service, and service-group profiles in demo.txt use the standard RADIUS attributes and vendor-specific RADIUS attributes (VSAs). The attributes are described in the *Cisco Subscriber Edge Services Manager and Subscriber Policy Engine Installation and Configuration Guide*. For example, the following profile from demo.txt is for the subscriber radiususer:

```
radiususer Password = "cisco"
        Service-Type = Framed-User,
        Account-Info = "Ainternet-blue",
        Account-Info = "Ninternet-blue",
        Account-Info = "Niptv",
        Account-Info = "Ngames",
        Account-Info = "Ndistlearn",
        Account-Info = "Ncorporate",
        Account-Info = "Nshop",
        Account-Info = "Nbanking",
        Account-Info = "Nvidconf"
        Account-Info = "Hhttp://www.spiderbait.com"
```

DESS Mode Functionality

To simulate DESS mode functionality (such as subscriber self-subscription, account management, and subaccount creation), the allowed subscriber profile attributes have been extended so that the NWSP web application can demonstrate DESS mode features:

- Different types of subscriber privileges (for service selection, service subscription, account management, and subaccount creation)
- Account attributes with X.500 information (title, given name, surname, and so on)
- Services and service groups that are available for subscription but not yet subscribed
- Parent account names for subaccounts

The extended set of attributes like other RADIUS attributes used by an SESM web application are read-only. However, NWSP in Demo mode does correctly simulate DESS mode functionality. For example, in Demo mode, if a subscriber adds or deletes a subscription, the list of subscribed services on the My Services page and in the service list are dynamically updated. The changes persist until the web server is stopped. The NWSP web application does not modify attributes in the Merit RADIUS file.

Note The extended set of subscriber profile attributes are allowed for Demo mode only. The extended set of attributes are not valid vendor-specific attributes (VSAs). They are not valid in RADIUS mode or LDAP mode and are not recognized by the SSG. They should not be added to the RADIUS dictionary.

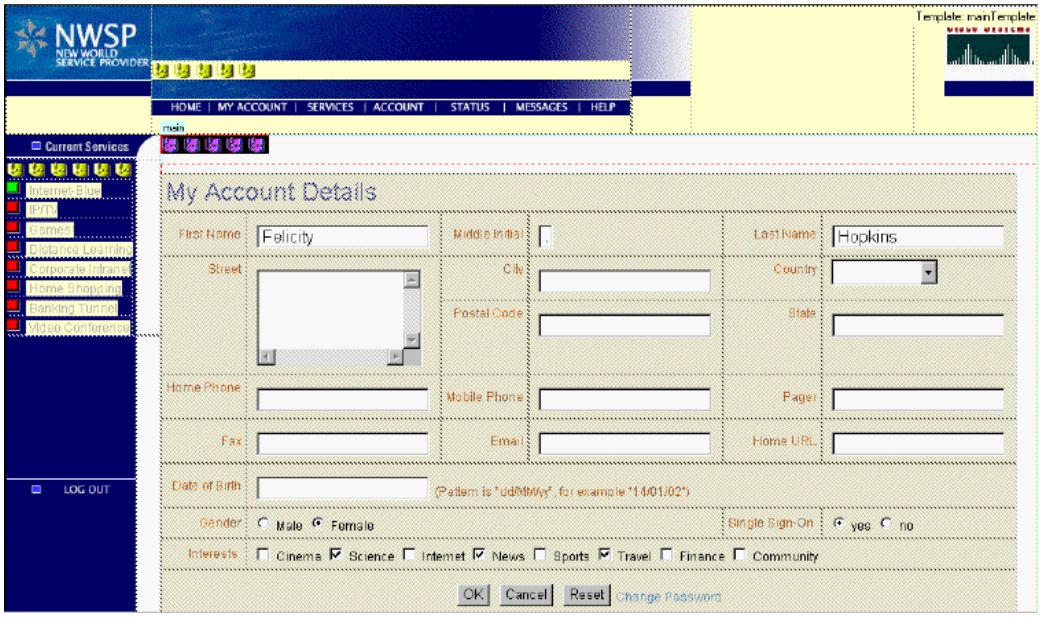
In NWSP, the demo.txt file contains two subscriber profiles for simulating DESS mode: ldapuser1 and ldapuser2. These subscriber profiles provide examples of the attribute extensions. For detailed information on the extended set of attributes for demonstrating DESS features, see the *Cisco Subscriber Edge Services Manager and Subscriber Policy Engine Installation and Configuration Guide*.

Dreamweaver UltraDev 4 Live Data Window

When you are developing a Cisco SESM web application, you can use the Live Data window feature of Dreamweaver UltraDev 4 to make changes to your JSP pages in a live data environment. Unlike the Document window, which uses placeholders for dynamic content on a JSP page, the Live Data window shows the actual dynamic content that is generated by the JSP page.

Figure 2-2 shows how myAccount.jsp is displayed in the Live Data window. To display dynamic data, UltraDev runs the JSP page on the web server before displaying it in the Live Data window. In Figure 2-2, notice that the Live Data window displays services in the service list and any subscriber data (for example, first name and last name) in the subscriber profile.

Figure 2-2 Live Data Window



Note Editing in a live data environment is available starting with Dreamweaver UltraDev 4.

To configure and use the Live Data window for the NWSP web application, perform the following steps. The procedure assumes that the web server is running on the local machine and listening for requests for the NWSP web application on port 8080.

Step 1 Start Dreamweaver UltraDev 4.

Step 2 In the **Site** menu, click **New Site**.

Step 3 In the **Category** list, click **Local Info** and specify the following:

Site Name: NWSP

Local Root Folder: *C:\SESM_location*\nwsp\docroot\ (In this procedure, *C:\SESM_location* is the location where the SESM software is installed.)

HTTP Address: http://localhost:8080

Step 4 In the **Category** list, click **Remote Info** and specify the following:

Access: Local/Network

Remote Folder: *C:\SESM_location*\nwsp\docroot\

Step 5 In the **Category** list, click **Application Server** and specify the following:

Server Model: JSP 1.0

Scripting Language: Java

Page Extension: .jsp

Access: Local/Network

Remote Folder: *C:\SESM_location*\nwsp\docroot\ (for example)

URL Prefix: http://localhost:8080/user=golduser/device=pc/locale=en/shape/

In URL Prefix, the values given after the port number are test decorator values that use URL-to-servlet mapping to specify the user, device, and locale. For information on using test values, see the "Using Test Decorators" section on page 2-14.

Note To use the test decorators specified in the URL Prefix, copy the contents of the web.dev.xml file to the end of the web.xml file used by the SESM web application, and then save the modified web.xml file. Restart the web server. The web.dev.xml file is located in the *install_dir*\nwsp\config directory.

Step 6 Click **OK**.

Step 7 To use the Live Data window to view a JSP page:

a. Double-click the name of the JSP page.

b. In the Document window that displays the JSP page, from the **View** menu, click **Live Data**.

UltraDev displays the actual dynamic content that is generated by the JSP page. In the Live Data window, you can edit the page's layout and JSP-page content. For information on editing in a live data environment, see *Using Dreamweaver UltraDev*, which is available on the web at:

http://www.macromedia.com/support/ultradev/documentation/using_ultradev_4.html

General Web Development Considerations

Some general web development considerations for a Cisco SESM web application include:

- We recommend the technique in which the web application redirects a `POST` request to a `GET` request. The Cisco SESM software has no mechanisms that support or prevent this technique.
- The decoration JSP pages and include files must not write and flush any characters to the `ServletOutputStream`. If either does, the forwarding HTTP request throws an `IllegalStateException`. This restriction is imposed by the servlet container.
- As is usual with any web application, all references to a web resource from a web page must be relative.

A web application can be deployed with a prefix, which is specified in the web.xml file. An absolute reference to another web resource within the same web application must include the web application prefix. However, the web application has no knowledge of the web application prefix. Therefore, references to web resources must be relative.

Debugging an SESM Web Application

If an SESM web application is not operating as expected, you can use the SESM logging utility to change the details of the information reported in the log files to diagnose a problem. For detailed information on logging and debugging mechanisms available for an SESM web application, see the *Cisco Subscriber Edge Services Manager and Subscriber Policy Engine Installation and Configuration Guide*.

The following hints may help with debugging an SESM web application:

- Most web servers can be set up so that the servlet code generated from JSP pages is saved and available for examination. With the Jetty web server, when the JSP pages are compiled, the servlet code is put in a temporary directory that is determined by the machine's Java Virtual Machine (JVM). Normally, the temporary directory is /tmp or /usr/tmp on UNIX, and C:\TEMP on Windows NT or Windows 2000. The servlet code can be examined in the temporary directory until it is deleted by the operating system's normal cleanup mechanism. For information on where servlet code is located with other web servers, refer to the documentation from your web server vendor.
- With the Internet Explorer browser, when an error occurs in the dynamic portion of a JSP page, Internet Explorer displays an alternative "friendly" (and less helpful) HTTP error message if the Show friendly HTTP error messages box is checked. To view server-generated error messages, make sure this box is not checked in the Advanced Tab for Internet Options, which you access through the Tools menu.
- Some SESM web application files such as web.xml and the properties files for resource bundles are read once when the web application is started. Changes to these files have no effect until the web application is stopped and restarted.

Using Test Decorators

The SESM software includes a number of Java servlets called *test decorators* that are very useful for development and testing of an SESM web application. The test decorators allow the SESM developer to set specific values for

- User name and password
- Locale language, country, and variant
- Other dimensions of the user shape such as brand and device

The web.dev.xml file has declarations for a number of test decorators preconfigured with some common values. For NWSP, the web.dev.xml file is located in the *install_dir*\nwsp\config directory. You can add to or modify the test values in web.dev.xml to meet your development and testing needs. For information on user-shape decoration and decorators, see the "SESM Software Concepts" section on page 3-6.

Note To use the test decorators, copy the contents of the web.dev.xml file to the end of the web.xml file used by the SESM web application, and then save the modified web.xml file. Restart the web server.

Each test decorator has an associated URL-to-servlet mapping that allows you to specify the test values in the URL when requesting an SESM web page. When the URL for a web application resource is preceded by URL-mapped test decorators, the corresponding test decorator servlets are invoked. Consider the following URL requesting myAccount—a control servlet for the My Account page:

http://localhost:8080/**user=golduser**/**device=pda**/**locale=de**/myAccount

When the preceding URL is used in demonstration mode, test decorator servlets are invoked and do the following:

- Authenticate the user name and password for golduser
- Set the `device` dimension of the user shape to `pda`
- Set the `locale` dimension of the user shape to `de` (Germany)

The following sections provide information on the test users, locales, and `device` and `brand` dimensions that are preconfigured in the web.xml file. For information on how you can modify and add to the set of test values and for information on the test decorator servlets, see Appendix B, "SESM Utility Servlets Quick Reference".

Test Users

Table 2-2 lists some of the test users, passwords, and URL mappings that are preconfigured in the web.dev.xml file. For each URL mapping, the `TestUserDecorator` servlet sets the user name and password to the values shown in the table. `TestUserDecorator` then automatically attempts to authenticate with the user name and password. In Demo mode, the SESM software authenticates against the subscriber profiles defined in the Merit RADIUS file demo.txt. The demo.txt file resides in the /config directory of the SESM web application.

Table 2-2 Test Users: Names and Password

User Name and Password	URL Mapping
User name: user1 Password: cisco	/user=user1/*
User name: user2 Password: cisco	/user=user2/*
User name: user31 Password: cisco	/user=user31/*
User name: user42 Password: cisco	/user=user42/*
User name: user43 Password: cisco	/user=user43/*
User name: user44 Password: cisco	/user=user44/*
User name: golduser Password: cisco	/user=golduser/*
User name: subgolduser Password: cisco	/user=subgolduser/*

Test Locales

Table 2-3 lists some of the test locales, associated values, and URL mappings that are preconfigured in the web.dev.xml file. For each URL mapping, the `LocaleDecorator` servlet sets the `locale` dimension to the value shown in the table.

Table 2-3 Test Locales

Locale	Value	URL Mapping
Australia	`au`	/locale=au/*
France	`fr`	/locale=fr/*
Germany	`de`	/locale=de/*

Table 2-3 Test Locales (continued)

Locale	Value	URL Mapping

Great Britain	`en`
	`gb`
	`uk`
		/locale=en/*
		/locale=gb/*
		/locale=uk/*

Italy	`it`	/locale=it/*
Portugal	`pt`	/locale=pt/*
Spain	`es`	/locale=es/*
Sweden	`se`	/locale=se/*
USA	`us`	/locale=us/*

Test Devices and Brands

Table 2-4 lists some of the test devices, brands, associated values, and URL mappings that are preconfigured in the web.dev.xml file. For each URL mapping, the `TestDimensionDecorator` servlet sets the `device` or `brand` dimension to the value shown in the table.

Table 2-4 Test Devices and Brands

Device or Brand	Value	URL Mapping
WAP device	`wap`	/device=wap/*
PDA device	`pda`	/device=pda/*
PC device	`pc`	/device=pc/*
XML device	`xml`	/device=xml/*
NWSP brand	`nwsp`	/brand=nwsp/*
Unstyled brand	`unstyled`	/brand=unstyled/*
Gold brand	`gold`	/brand=gold/*
Silver brand	`silver`	/brand=silver/*
Bronze brand	`bronze`	/brand=bronze/*
No brand	an empty string	/brand=/*

Using Device Simulators for WAP and WML

WAP phone simulators are very useful for development and testing an SESM web application that generates WAP content using Wireless Markup Language (WML). The sample WAP web application is designed for providing content to a subscriber who is using a WAP phone. WAP phone simulators allow you to view WAP content by using the HTTP protocol to access the JSP pages of an SESM web application. Figure 2-3 shows the home page and service list of the WAP web application as it appears on a Nokia simulator.

Figure 2-3 WAP Phone Simulator

When the HTTP protocol mode is used to access an SESM web application, you request pages from an SESM web application in the usual manner. For example, to log on to SESM, you specify the following URL: http://*web_server*:8080, where *web_server* is the IP address of the Jetty web server.

The NWSP and WAP web applications recognize many WAP phone simulators, including the Nokia Mobile Internet Toolkit simulator, the Openwave UP.Simulator (version4.0 and 4.1), and the WinWAP Pro simulator. The NWSP and WAP web applications set the `device` dimension of the user shape to `wap` when these simulators are the client device.

WAP phone simulators that are available on the web include:

- Nokia Mobile Internet Toolkit contains two simulators and is available at Forum Nokia:

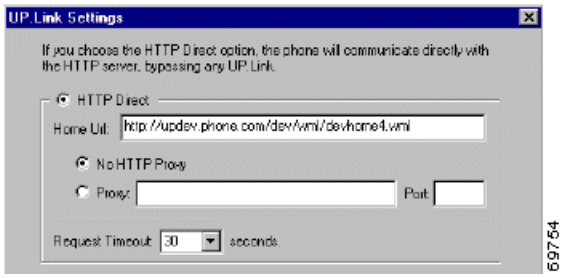
  http://www.forum.nokia.com/

- UP.SDK for WML from Openwave Systems Inc. contains a simulator and is available at:

  http://developer.openwave.com

To use the UP.Simulator and the HTTP protocol to access an SESM web application, the UP.Simulator must be configured to use HTTP Direct as the UP.Link Setting. To do this, start the UP.Simulator, click UP.Link Settings from the Settings menu, and then click HTTP Direct in the UP.Link Settings dialog box (Figure 2-4).

Figure 2-4 UP.Link Settings for the UP.Simulator



Managing an SESM Web Site

Dreamweaver has a number of features that may be useful for developing an SESM web application. For example, if a Dreamweaver template is modified, you can automatically update all files in a web site that use the template. To use some Dreamweaver features, you must define a web site in Dreamweaver.

To define an SESM web application as a site, do the following:

Step 1 From the **Dreamweaver Site** menu, choose **New Site** and specify the required information.

Step 2 In the **Local Info** dialog box, specify the */webapp/docroot* directory of the SESM web application as the Local Root Folder.

Step 3 In the **Site Map Layout** dialog box, specify home.jsp as the Home Page.

After the site is defined in this manner, Dreamweaver creates a useful Site Files view. However, because home.jsp is designed to contain few links to other JSP pages, the Site Map view is not very useful.

For detailed information on setting up and managing a site, refer to the Dreamweaver documentation.

The Dreamweaver Preview in Browser feature is of limited use with an SESM web application. For example, in the NWSP web application, most JSP pages require some form of input. Without the input, the web browser displays an error page. If you are using Dreamweaver UltraDev 4, the Live Data window feature is an excellent alternative to the Preview in Browser feature. The Live Data window allows you to view dynamic data in a sample SESM web application. For information on configuring the Live Data window, see the "Dreamweaver UltraDev 4 Live Data Window" section on page 2-11.