Chapter 3
Customizing the public access interface

This chapter provides you with an overview of the public access interface and shows you how to customize it.

Overview

The public access interface is the sequence of web pages that customers use to log in, log out, and view the status of their wireless connections to the public access network.

The MSC enables you to tailor the public access interface web pages to provide a customized look-and-feel for your site. Web pages can be automatically updated using a RADIUS server, enabling you to manage multiple units effortlessly.

Note: *Customers using PDAs that support a single browser window will have difficulty using the public access interface in its standard configuration. For information on how to correct this problem, see* "Supporting PDAs" on page 68.

Common configuration tasks

The following table lists some common configuration tasks and indicates where to find more information.

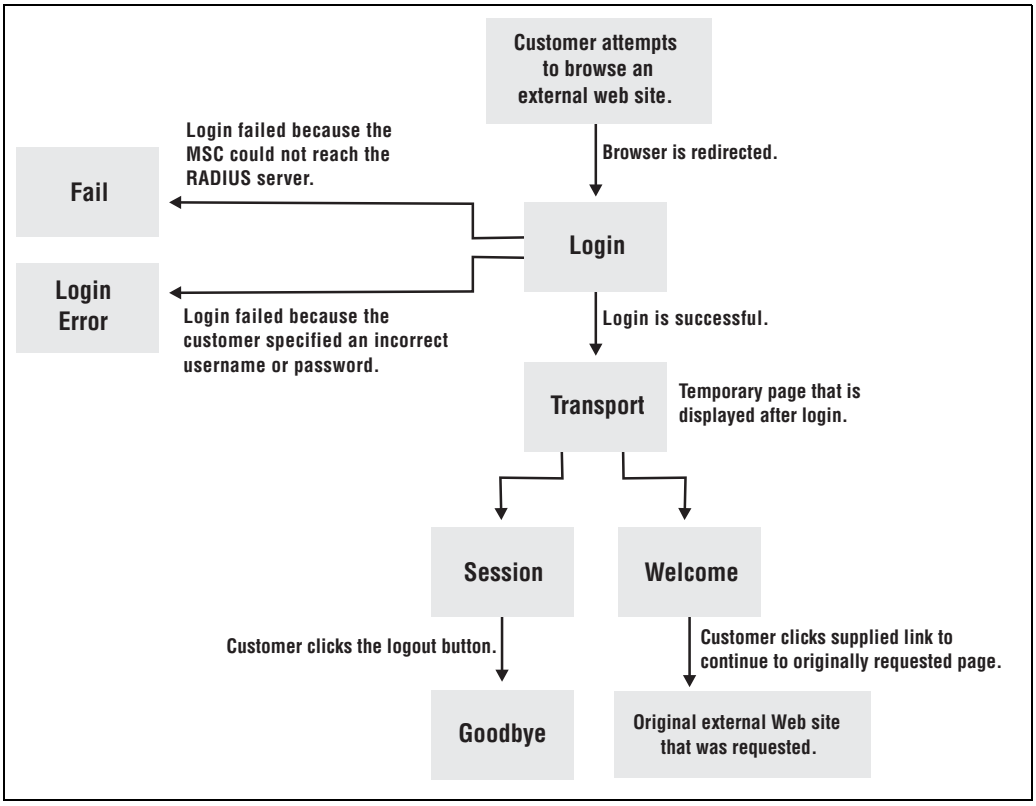
Task	For instructions
Changing the Login page and logo	See page 62
Hosting the login page on your own web server	See page 69
Displaying custom Welcome or Goodbye pages	See page 67
Delivering custom content based on a customer's location in the network	See page 67
Supporting PDAs	See page 68
Restricting customer logins based on their location in the network	See page 75

Site map

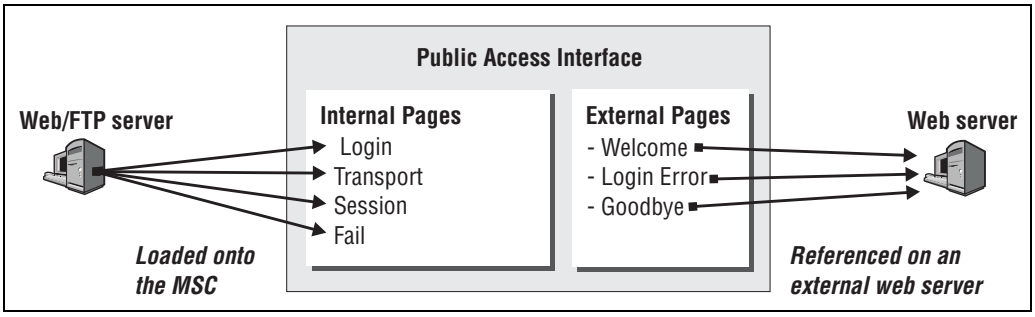
This section describes how the public access interface is structured and provides an overview of each component.

Structure

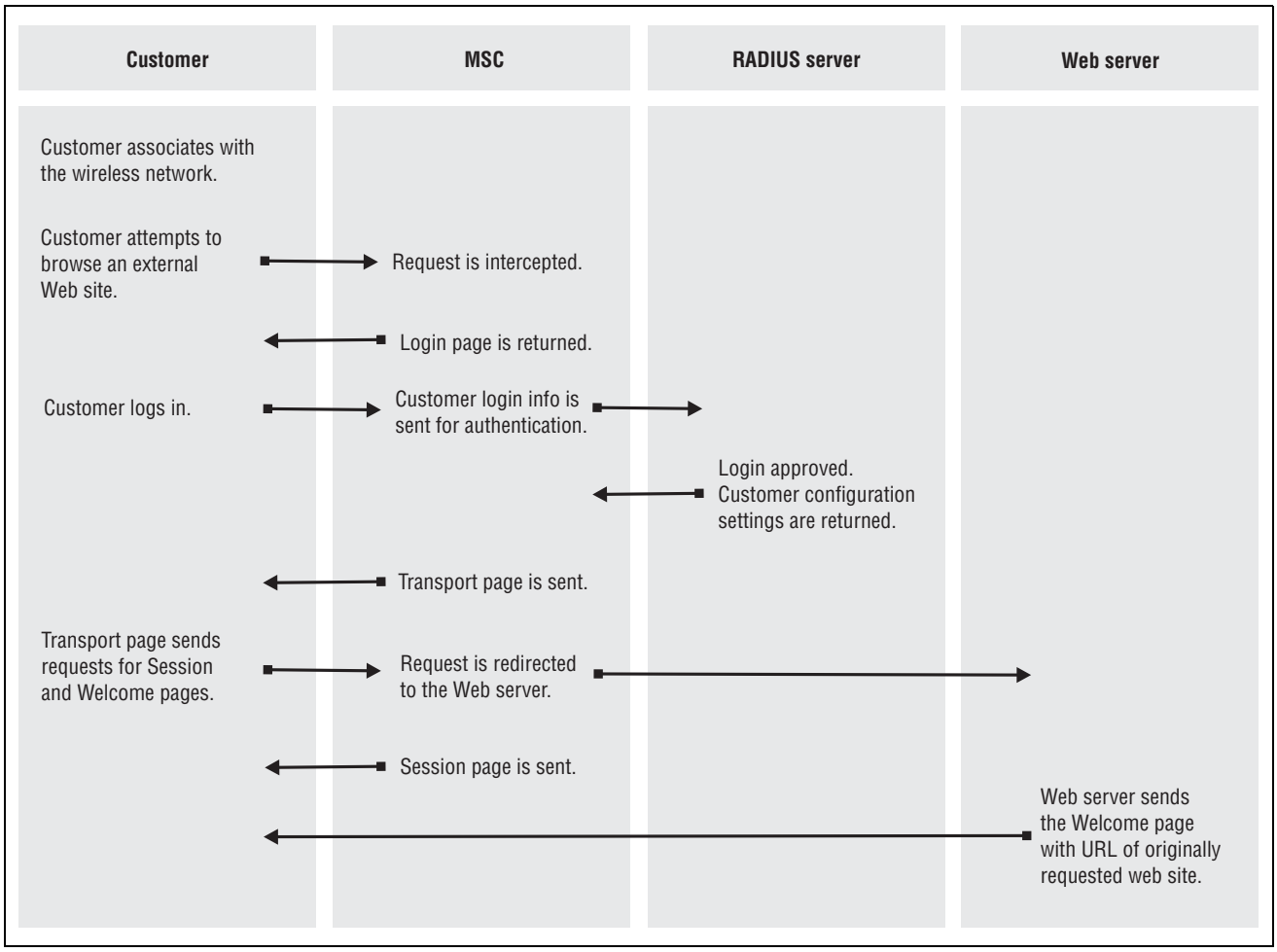
The default public access interface comprises seven pages and is structured as follows:



Pages are categorized into two groups: internal pages and external pages.



The following diagram shows the sequence of events that occur when a customer attempts to browse an external web site. This example assumes that the default public access setup is being used with a RADIUS server.



Internal pages

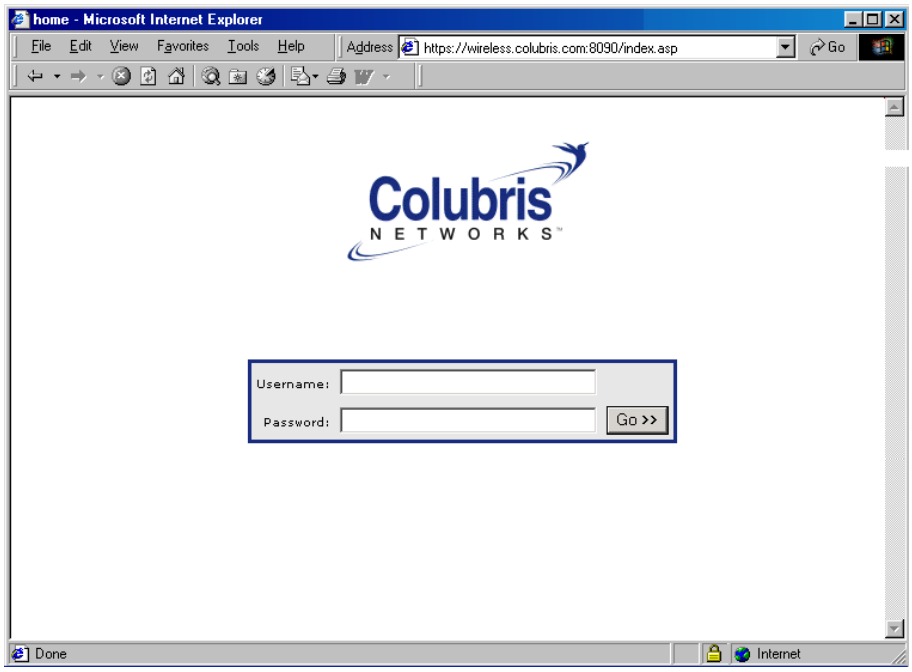
The internal pages reside on the MSC. You can use the default pages supplied with the MSC, or you can replace them with customized pages of your own design.

Login page

The Login page contains a single graphic element suitable for a logo or other identifying element and two fields: username and password.

Note: *Customers using 802.1x/WPA are automatically logged in and do not see the Login page.*

The default Login page is shown below:



You can also create a remote login page that resides on an external web server and is not downloaded to the MSC. For details see "Using a remote login page" on page 69.

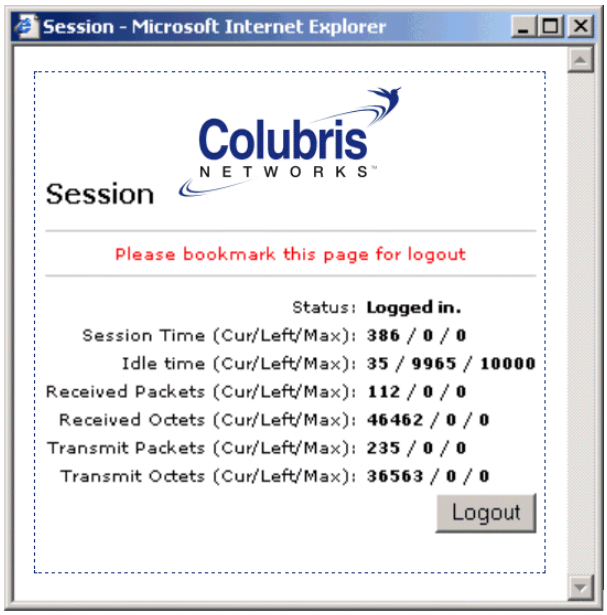
Transport page

The **Transport** page appears briefly and spawns the **Session** and **Welcome** pages.

Session page

The **Session** page shows usage statistics for the session, as well as the logout button that the customer clicks to terminate the session.

The default **Session** page is shown below:



Managing the Session page

The **Session** page opens automatically after the customer logs in. By default it contains the logout button. Without the **Session** page the customer cannot log out. The following URL can be used to reopen the session page if a customer accidentally closes it.

`http://MSC_hostname:port/session.asp.`

For example:

`http://wireless.colubris.com:8080/session.asp`

Launching the Session page from the Welcome page

You can embed the following URL on the **Welcome** page to dynamically link to the **Session** page:

`<a href="http://MSC_hostname:port/session.asp">Session page</a>`

Forcing a logout

You can force a logout with this URL:

`http://MSC_hostname:port/goform/HtmlLogout`

For example:

`http://wireless.colubris.com:8080/goform/HtmlLogout`

Customers with PDAs

Customers using PDAs that support only a single browser window never see the session page. This makes it impossible for them to log out. For information on how to correct this problem, see "Supporting PDAs" on page 68.

Fail page

The Fail page appears if the MSC cannot contact the RADIUS server to authenticate a customer.

Note: *Customers using 802.1x/WPA are automatically logged in and do not see the **Fail** page.*

External pages

The external pages are hosted by an external web server. The MSC redirects customers to these pages as required.

The MSC can be configured to use a different external web page for each customer if required.

Welcome page

The **Welcome** page appears after the customer has successfully logged in and can be used to provide information about the public access network and its options. The Welcome page also includes a link to the page that was originally requested by the customer. If the MSC cannot reach the custom URL specified for the **Welcome** page, or if a custom URL is not defined, it jumps directly to the page originally requested by the customer.

Goodbye page

The **Goodbye** page acknowledges a customer logout.

Login error page

The **Login error** page appears if the customer cannot be authenticated. The reason is shown on the page. You can customize the messages on this page by editing the file **messages.txt**. See “Customizing error messages” on page 64.

Remote Login page

Instead of using the internal Login page you can create a remote login page that resides on an external web server. For details see “Using a remote login page” on page 69.

Customizing the internal pages

This section explains how to create new internal pages, as well as how to edit the shared image file (logo) and the message file.

Creating new internal pages

To create new internal pages, use the sample pages provided on the Colubris Networks documentation CD as a starting point. See page 64 for examples.

Important: *Do not create new pages by saving an internal page while viewing it within your web browser. If you do so, the server-side code is removed, and the resulting pages will not work.*

The internal pages use a number of Colubris-specific ASP functions to display status information. You can also use these functions to enhance your custom pages. Descriptions of these functions start on page 78.

Important restrictions

Because the internal pages must be loaded onto the MSC, the following restrictions apply to their construction.

- **You must specify a URL for ALL internal pages, even if you want to change only one page. Simply use copies of the standard internal pages for the pages you do not want to change.**
- Do not alter the ID tags “<!-- Colubris -->” & “<!-- Custom -->” located at the top of the page.
- Do not alter any JavaScript code, except for the **Session** window parameters *width* and *height*.
- Only one image can be included on these pages. It must be a *.gif* file, and Colubris recommends that the file size be less than 20K. This same image file is shared by all pages and must be resident on the MSC. For instructions on how to change the image, see “Examples” on page 64.
- Do not alter any occurrences of “Get...();” or “GetWelcomeURL();”
- Do not alter any form elements. Leave intact all names and values.
- Do not change the filename extensions of the internal pages.

Loading new internal pages

To load new internal pages, you must define the URLs where the MSC can download them using a service controller attribute. The attribute can be defined in the RADIUS account for the MSC (if you use a RADIUS server) or they can be locally configured.

See the following topics for more information:

- “Configuring the public access network” on page 22.
- “Service controller attributes” on page 29.

Colubris-AVPair value strings

The following table presents the Colubris-AVPair value strings used for customizing the internal pages.

Internal page	Colubris-AVPair value string	Notes
Login	`login-page=`*`URL_of_page`*	Required. (Unless a remote login page is being used as explained on page 69).
Transport	`transport-page=`*`URL_of_page`*	Required.
Session	`session-page=`*`URL_of_page`*	Required.
Fail	`fail-page=`*`URL_of_page`*	Required.
Re-usable image	`logo=`*`URL_of_gif_file`*	Required. This image is shared by all pages.
Error messages	`messages=`*`URL_of_text_file`*	Optional. These messages appear when various error conditions occur.

Important: ***The maximum length of any internal page URL is 512 characters. If this is exceeded (when using placeholders for example), the URL is truncated. Colubris therefore recommends that you specify the most-important placeholders first.***

Important: ***The internal pages can only be changed as a group.*** *You cannot, for example, just use the login-page string in a RADIUS profile. You must use all required items. This means that the minimum set you can specify is as follows:*

```
login-page= URL_of_page
transport-page= URL_of_page
session-page= URL_of_page
fail-page= URL_of_page
logo= URL_of_gif_file
```

Placeholders

The following optional placeholders can be appended to the Colubris-AVPair value strings for the internal pages. These placeholders are not available in local mode.

Placeholder	Description
`%n`	Returns the NAS ID assigned to the MSC. By default, this is the unit's serial number.
`%s`	Returns the RADIUS login name assigned to the MSC. By default, this is the unit's serial number.
`%i`	Returns the domain name assigned to the MSC's Internet port.
`%a`	Returns the IP address of the MSC's interface that is sending the authentication request.

Examples

These examples show how to accomplish common customization tasks using the sample files on the Colubris Networks documentation CD as a starting point. To retrieve the sample files, do the following:

1. Select **Public Access Examples** on the main menu.
2. Select **Sample HTML Files for V5.1**.
3. Copy **Internal_Pages.zip** to a folder on your computer and extract the files.

Changing the login page and logo

1. Create a folder called **newpages** on your web sever.
2. Create a file called **logo.gif** that contains your logo and place it in the **newpages** folder.
3. Copy the following files from **Internal_Pages.zip** and place them in the **newpages** folder.
   - login.html
   - transport.html
   - session.html
   - fail.html
4. Edit **login.html** to customize it for your site.
5. Add the following entries to the **Configured attributes** table on the **Public access > Attributes** page. (You can also define these attributes in the RADIUS profile for the MSC if you are using a RADIUS server.)

```
login-page=web_server_URL/newpages/login.html

transport-page=web_server_URL/newpages/transport.html

session-page=web_server_URL/newpages/session.html

fail-page=web_server_URL/newpages/fail.html

service-announcement-page=web_server_URL/newpages/service-
announcement.html

logo=web_server_URL/newpages/logo.gif
```

Customizing error messages

Several of the internal pages use the functions GetAuthenticationErrorMessage() and GetSessionStateMessage() to return a string from the file **message.txt**. You can customize the messages in this file for your installation as follows:

1. Create a folder called **newpages** on your web sever.
2. Copy the file messages.txt from **Internal_Pages.zip** and place it in the **newpages** folder.
3. Edit **messages.txt** with an ASCII editor. Customize the messages to suit your installation.
4. Add the following entry to the **Configured attributes** table on the **Public access > Attributes** page. (You can also define this attribute in the RADIUS profile for the MSC if you are using a RADIUS server.)

```
messages=web_server_URL/newpages/messages.txt
```

Customizing the external pages

This section explains how to customize the three external pages: **Welcome, Login error,** and **Goodbye.**

Creating new external pages

Unlike the internal pages, the external pages do not have any restrictions on their construction since they reside on a third-party server. See page 67 for examples

Activating new external pages

To activate new external pages, you must define their URLs using the Colubris-AVPair value string when you create a RADIUS profile for the MSC or a customer. See the MSC administrator's guide for information on how to create RADIUS profiles.

When the MSC authenticates itself, or a customer, it retrieves the URLs for the custom pages, then automatically redirects customers to them when required.

Note: *The MSC maintains a separate copy of the URLs for external pages for each customer. This means it is possible to provide different pages for each customer. See "Displaying custom welcome and goodbye pages" on page 67.*

The following table presents the Colubris-AVPair value strings used for customizing the external pages.

Attribute	Notes
`login-err-url=`*`URL_of_page[placeholder]`*	Access to the web server hosting this page must be granted to all unauthenticated customers. Do this with an appropriate access list definition. (Customers can see this page *before* they are logged in.)
`welcome-url=`*`URL_of_page[placeholder]`*	The customer is authenticated, so the welcome page can be located on any URL reachable by the customer.
`goodbye-url=`*`URL_of_page[placeholder]`*	Access to the web server hosting this page must be granted to all unauthenticated customers. Do this with an appropriate access list definition. (Customers see this page *after* they are logged out.)

Important: ***The maximum length of any external page URL is 512 characters. If this is exceeded (when using placeholders for example), the URL is truncated. Colubris therefore recommends that you specify the most-important placeholders first.***

An important feature of the external pages is that they make it easy to deliver a unique experience for each customer. By appending the following optional placeholders to the Colubris-AVPair value strings for the external pages, you can pass important information to the web server. Server-side code can process this information to generate custom pages on-the-fly.

Placeholder	Description
`%c`	Returns the IP address of the customer's computer.

Placeholder	Description
%d	Returns the WISPr location-ID. Supported for login-url only.
%e	Returns the WISPr location-Name. Supported for login-url only.
%l	Returns the URL on the MSC where customer login information should be posted for authentication. This option is used with the remote login page feature. By default, this value is URL encoded. (To enable/disable URL encoding, set the value of **url-encode** in the **<ACCESS-CONTROLLER>** section in the configuration file.)
%n	Returns the NAS ID assigned to the MSC. By default, this is the unit's serial number. Not supported in local mode.
%s	Returns the RADIUS login name assigned to the MSC. By default, this is the unit's serial number.
%u	Returns the login name of the customer.
%o	Returns the original URL requested by the customer. By default, this value is URL encoded. (To enable/disable URL encoding, set the value of **url-encode** in the **<ACCESS-CONTROLLER>** section in the configuration file.)
%i	Returns the domain name assigned to the MSC's Internet port.
%p	Returns the IP port number on the MSC where customer login information should be posted for authentication.
%a	Returns the IP address of the MSC's interface that is sending the authentication request.
%E	When the location-aware feature is enabled, returns the ESSID of the wireless access point the customer is associated with.
%P	When the location-aware feature is enabled, returns the wireless mode ("ieee802.11a", "ieee802.11b", "ieee802.11g") the customer is using to communicate with the access point.
%G	When the location-aware feature is enabled, returns the group name of the wireless access point the customer is associated with.
%C	When the location-aware feature is enabled, returns the Called-station-id content for the wireless access point the customer is associated with.
%r	Returns the string sent by the RADIUS server when an authentication request fails. The RADIUS server must be configured to support this feature. The information contained in the returned string depends on the configuration of the RADIUS server.
%m	Returns the MAC address of the client station that is being authenticated.
%v	Returns the VLAN assigned to the client station at the MSC's ingress (LAN port).

Customization examples

These examples show how to accomplish common customization tasks using the sample files on the Colubris Networks documentation CD as a starting point. To retrieve the sample files, do the following:

1. Select **Public Access Examples** on the main menu.
2. Select **Sample HTML Files for V5.1**.
3. Copy **External_Pages.zip** to a folder on your computer and extract the files.

Displaying custom welcome and goodbye pages

This example shows how to display unique welcome and goodbye pages for specific customers or groups of customers.

For this example, assume you have two sets of customers: basic and premium. To distinguish the two groups, you have set up the customer accounts on the RADIUS server accordingly. (Perhaps you are using access lists to restrict each group to a different section of the public network as described on page 36).

1. Create the following two folders on your web sever: **basic** and **premium**.
2. Copy **welcome.html** and **goodbye.html** from **External_Pages.zipl** into each folder.
3. Customize **welcome.html** and **goodbye.html** in each folder for each set of customers.
4. Add the following entry to the RADIUS profile for the basic customers.

   ```
   welcome-url=web_server_URL/basic/welcome.html
   goodbye-url=web_server_URL/basic/goodbye.html
   ```

5. Add the following entry to the RADIUS profile for the premium customers.

   ```
   welcome-url=web_server_URL/premium/welcome.html
   goodbye-url=web_server_URL/premium/goodbye.html
   ```

6. Add the following entry to the RADIUS profile for the MSC. This gives all unauthenticated users access to the web server hosting the goodbye page.

   ```
   access-list=loginserver,ACCEPT,tcp,web_server_IP_address,port_number
   ```

Delivering dynamically generated content

Another way to generate custom pages is to add placeholders in the URLs for the custom external pages and then use server-side scripting to dynamically create the pages. This method provides a powerful mechanism to automatically generate completely customized pages on a per-user basis. Rather than designing one or more static pages, as in the previous example, the custom pages in this example can be built on-the-fly based on customer preferences stored in a central database, or based on a customer's location within the network.

For example, if you want to generate a custom welcome page for each customer:

1. Add the following entry to the RADIUS profile for the MSC.

   ```
   welcome-url=web_server_URL/premium/welcome.html
   ?loginname=%u&IPaddress=%i
   ```

2. Create a server-side script to retrieve the customer's login name (%u) and the MSC's IP address or domain name (%u). The script can use this information to then display a custom page based on customer's preferences (stored in the server's database) and the customer's location within the wireless network.

Supporting PDAs

Customers using PDAs that only support a single browser window will have difficulty using the public access interface in its standard configuration.

Once a customer logs in to the public access interface, two web pages are sent to their browser: the Welcome page and the Session page.

The Session page contains a logout button. Customers who are unable to view this page will not be able to log out.

To solve the problem, modify the Welcome page to include a logout button.

1. Create a folder called **PDAcustomers** on your web sever.
2. Copy **welcome.html** and **goodbye.html** from **External_Pages.zip** into this folder.
3. Edit **welcome.html** to include a logout link with the target:

   `http://`*`MSC_ip`*`:`*`port`*`/goform/HtmlLogout.`

   For example:

   `http://192.168.1.1:8080/goform/HtmlLogout.`

   Add a warning to this page that tells PDA customers to bookmark the Welcome page so that they can logout.
4. Add the following entry to the RADIUS profile for all PDA customers.

   `welcome-url=`*`web_server_URL`*`/PDAcustomers/welcome.html`

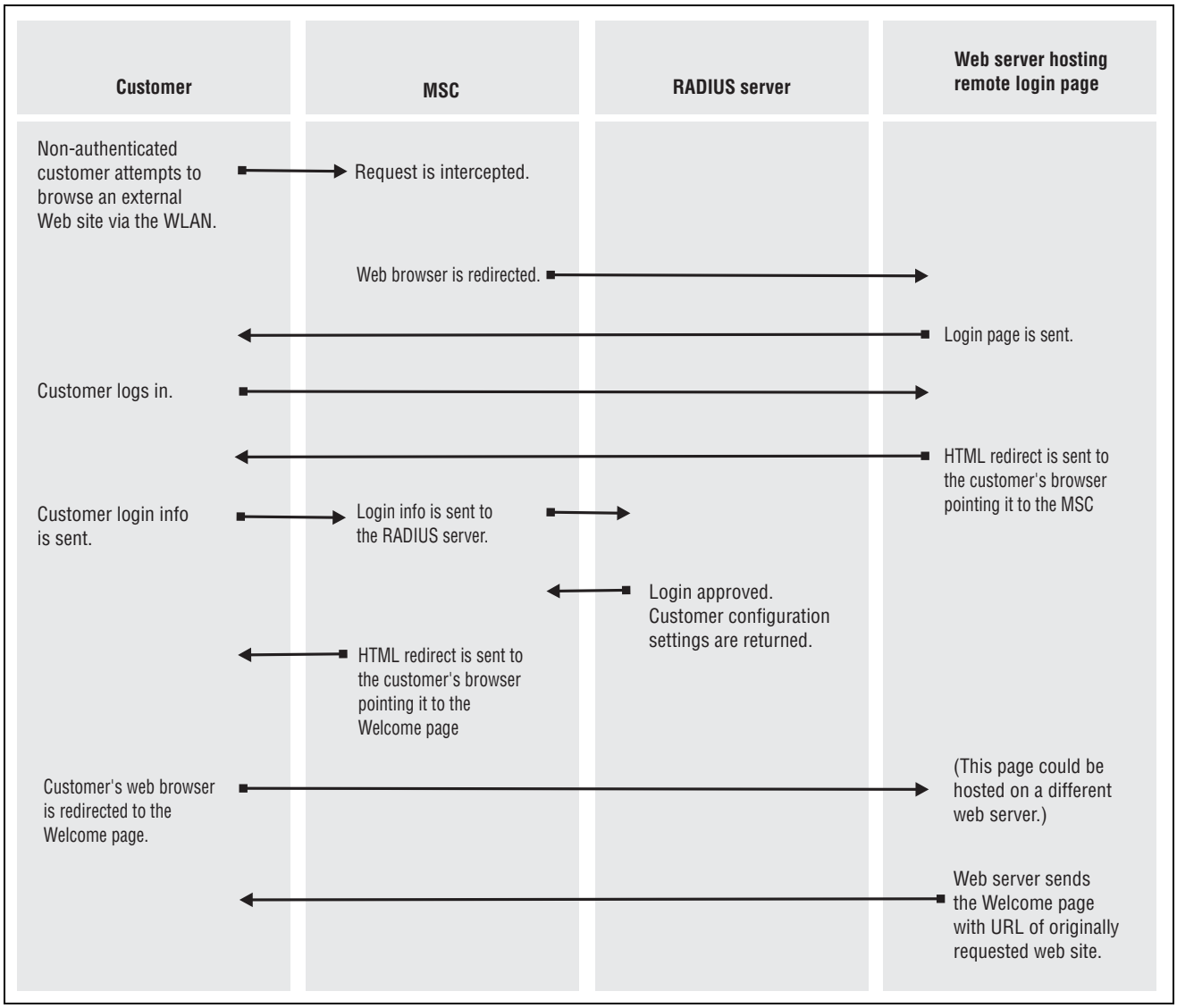
Using a remote login page

The MSC provides an option that enables you to redirect customers to a remote server to log in to the public access interface instead of using the internal login page. Hosting the login page on a remote server means that the login page is completely customizable. You are not bound by the limits imposed by loading a login page onto the MSC.

How it works

Although the remote login page feature enables you to host the public access login page on a remote web server, authentication of customers is still performed by the MSC through a RADIUS server or using the local user list. To accomplish this, the remote web server must send customer login information back to the MSC. There are two ways this can be done: basic remote login (as described in this section), or by using the NOC-based authentication feature (described in Chapter 4).

The following diagram shows the sequence of events for a typical customer session when using a remote login page and a RADIUS server for authentication.



Activating a remote login page

To activate a remote login page, you must define the URL where the MSC can redirect login requests using a service controller attribute. The attribute can be defined in the RADIUS account for the MSC (if you are using a RADIUS server) or it can be locally configured.

Use the following Colubris-AVPair value string:

`login-url=`*`URL_of_the_page`* `[`*`placeholder`*`]`

Where:

Parameter	Colubris-AVPair value string
URL_of_the_page	URL of the remote login page. Access to the web server hosting this page must be granted to all unauthenticated customers. Do this with an appropriate access list definition.

The following placeholders can be added to the login-url string.

Placeholder	Description
%c	Returns the IP address of the customer's computer.
%d	Returns the WISPr location-ID. Supported for login-url only.
%e	Returns the WISPr location-Name. Supported for login-url only.
%l	Returns the URL on the MSC where customer login information should be posted for authentication. By default, this value is URL encoded. (To enable/disable URL encoding, set the value of **url-encode** in the **<ACCESS-CONTROLLER>** section in the configuration file.)
%n	Returns the NAS ID assigned to the MSC. By default, this is the unit's serial number. Not supported in local mode.
%s	Returns the RADIUS login name assigned to the MSC. By default, this is the unit's serial number. Not supported in local mode.
%o	Returns the original URL requested by the customer. By default, this value is URL encoded. (To enable/disable URL encoding, set the value of **url-encode** in the **<ACCESS-CONTROLLER>** section in the configuration file.)
%i	Returns the domain name assigned to the MSC's Internet port.
%p	Returns the port number on the MSC where customer login information should be posted to for authentication.
%a	Returns the IP address of the MSC's interface that is sending the authentication request.
%E	When the location-aware feature is enabled, returns the ESSID of the wireless access point the customer is associated with.
%P	When the location-aware feature is enabled, returns the wireless mode ("ieee802.11a", "ieee802.11b", "ieee802.11g") the customer is using to communicate with the access point.
%G	When the location-aware feature is enabled, returns the group name of the wireless access point the customer is associated with.
%C	When the location-aware feature is enabled, returns the Called-station-id content for the wireless access point the customer is associated with.

Placeholder	Description
%r	Returns the string sent by the RADIUS server when an authentication request fails. The RADIUS server must be configured to support this feature. The information contained in the returned string depends on the configuration of the RADIUS server.
%m	Returns the MAC address of the wireless/wired client station that is being authenticated.
%v	Returns the VLAN assigned to the client station at the MSC's ingress (LAN port).

Important: ***The maximum length of the remote login page URL is 512 characters. If this is exceeded (when using placeholders for example), the URL is truncated. Colubris therefore recommends that you specify the most-important placeholders first.***

Important: ***To use the remote login page, you must also define a complete set of internal pages (except for login.html), which includes:***

```
transport-page=URL_of_page
session-page=URL_of_page
fail-page=URL_of_page
logo=URL_of_gif_file
```

Security issues

- Colubris recommends that the web server hosting the remote login page be secured with SSL (requires an SSL certificate from a well-known certificate authority), to ensure that customer logins are secure. Without SSL security, logins are exposed and may be compromised, enabling fraudulent use of the network.
- Communications between the customer's browser and the MSC is always SSL-based. The default certificate on the MSC generates a warning on the customer's browser unless replaced with a certificate signed by a well-known certificate authority.

Example

This example uses the sample files on the Colubris Networks documentation CD as a starting point. To retrieve the sample files, do the following:

1. Select **Public Access Examples** on the main menu.
2. Select **Sample HTML Files for V5.1**.
3. Copy **Internal_Pages.zip** to a folder on your computer and extract the files.

To enable a basic remote login page, do the following:

1. Create the following folder on your web sever: **newlogin**
2. Copy the following files from **Internal_Pages.zip** and place them in the **newlogin** folder.
   - login.html
   - transport.html
   - session.html
   - fail.html
   - logo.gif

3. Add the following entries to the **Configured attributes** table on the **Public access > Attributes** page. (You can also define these attributes in the RADIUS profile for the MSC if you are using a RADIUS server.)

```
login-url=web_server_URL/newlogin/login.html?loginurl=%l
transport-page=web_server_URL/newlogin/transport.html
session-page=web_server_URL/newlogin/session.html
fail-page=web_server_URL/newlogin/fail.html
logo=web server URL/newlogin/logo.gif
access-list=loginserver,ACCEPT,tcp,web_server_IP_address
use-access-list=loginserver
```

4. Customize **login.html** to accept username and password information from customers and then send it to the MSC. You can use code similar to the following example to redirect the customer's web browser to the login URL on the MSC for authentication:

```
<form action="https://wireless.colubris.com:8090/goform/
HtmlLoginRequest" method="POST">
```

For more flexibility, the remote login page should be written using a server-side scripting language such as ASP, PHP, or PERL. This enables the remote login page to take advantage of the placeholders that may have been defined in the login-url section of the RADIUS profile.

WISPr support

The public access interface provides support for WISPr (Wireless Internet Service Project Roaming) using WISPr and Colubris vendor-specific attributes.

WISPr vendor-specific attributes

Colubris Networks supports three Wi-Fi Alliance vendor-specific attributes for Access Request and Accounting Request. These attributes are:

Location-Name

- SMI network management private enterprise code = 14122
- Vendor-specific attribute type number = 2
- Attribute type = string

Location-ID

- SMI network management private enterprise code = 14122
- Vendor-specific attribute type number = 1
- Attribute type = string

Logoff-url

- SMI network management private enterprise code = 14122
- Vendor-specific attribute type number = 3
- Attribute type = string

Colubris Networks vendor-specific attributes

WISPr login URL

This attribute lets you define the location of the WISPr login page. The MSC automatically redirects customers with WISPr-compatible wireless client software to this page. To customize the redirection use the WISPr redirect page attribute.

Use the following Colubris-AVPair value string:

`wispr-login-url=`*`URL_of_page`*

Where:

Parameter	Description
`URL_of_page`	URL of the WISPr login page.

WISPr abort login URL

This attribute lets you define the destination where the WISPr abort login will be POSTed.

Use the following Colubris-AVPair value string:

`wispr-abort-login-url=`*`URL_of_page`*

Where:

Parameter	Description
URL_of_page	URL where to POST the WISPr abort login.

WISPr redirect page

This attribute lets you define the location of the WISPr redirect page. Use this page to customize the code that the MSC includes in the HTTP redirect sent to a customer's browser.

`redirect-page=`*`URL_of_page`*

Where:

Parameter	Description
URL_of_page	URL of the page containing code to use for WISPr redirect.

If this attribute is not defined the following code is used by default:

```
<!-- Colubris -->
<!-- Default -->
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<!--iPass
<WISPAccessGatewayParam>
  <Redirect>
    <MessageType>100</MessageType>
    <ResponseCode><% iPassGetRedirectResponseCode(); %></ResponseCode>
    <AccessProcedure><% iPassGetAccessProcedure(); %></AccessProcedure>
    <LocationName><% iPassGetLocationName(); %></LocationName>
    <AccessLocation><% iPassGetAccessLocation(); %></AccessLocation>
    <LoginURL><% iPassGetLoginUrl(); %></LoginURL>
    <AbortLoginURL><% iPassGetAbortLoginUrl(); %></AbortLoginURL>
  </Redirect>
</WISPAccessGatewayParam>
-->
<!-- Boingo
<smartClient>
  <page>
    <login>
      <login_url><% BoingoGetLoginUrl(); %></login_url>
    </login>
  </page>
</smartClient>
-->
</html>
```

The source code for this page is available in the file **redirect.html** which is available on the Colubris Networks documentation CD. To retrieve this file, do the following:

1. Select **Public Access Examples** on the main menu.
2. Select **Sample HTML Files for V5.1**.
3. Copy **Internal_Pages.zip** to a folder on your computer and extract the files.

Location-aware authentication

This feature enables you to control logins to the public access network based on the wireless access point a customer is associated with. Once authenticated, this feature is also used to monitor and control roaming to other access points in the network.

How it works

Location-aware is automatically enabled when a VSC is set to **provide access control**. When enabled, the location-aware feature causes the MSC to return location-specific information for RADIUS-authenticated customers. This information is returned:

- when the customer logs in
- each time the customer roams to a new access point or switches SSIDs on the same access point (which causes the customer to be re-authenticated)

Note: *Due to security constraints in 802.1x client software, customers cannot automatically be re-authenticated when roaming to a new access point. Therefore, location-aware information cannot be returned when these customer's roam.*

Returned information

The MSC can return the following attributes in the RADIUS access request for all customer authentications (whether initial login or re-authentication due to roaming).

- Called-station-ID (Standard RADIUS attribute)
- Colubris-specific attribute: SSID
- Colubris-specific attribute: GROUP

Note: *When re-authenticating customers, the returned RADIUS attribute Service-Type is set to 8744 (decimal).*

Called-Station-ID value

By default, this is the MAC address of the wireless port (radio) the customer is associated with. This is the MAC address of the **wvlan0** or **wvlan1** interface in IEEE format as displayed by **Tools > System Tools > Interface info.**

If required, the MSC can return other values for this attribute by setting the **Called-Station-Id content** on a per-VSC basis. The other available options are:

- SSID: SSID of the access point the customer is associated with.
- GROUP: Group name of the access point the customer is associated with.

Note: *If the customer is connected via a wired connection, the value returned is the MAC address of the MSC's wireless/LAN port. To use the MAC address of the Internet port, you must edit the config file and change the setting of* ***radius-called-station-id-port*** *to* ***WAN*** *in the <ACCESS-CONTROLLER> section.*

Colubris-specific attribute: SSID

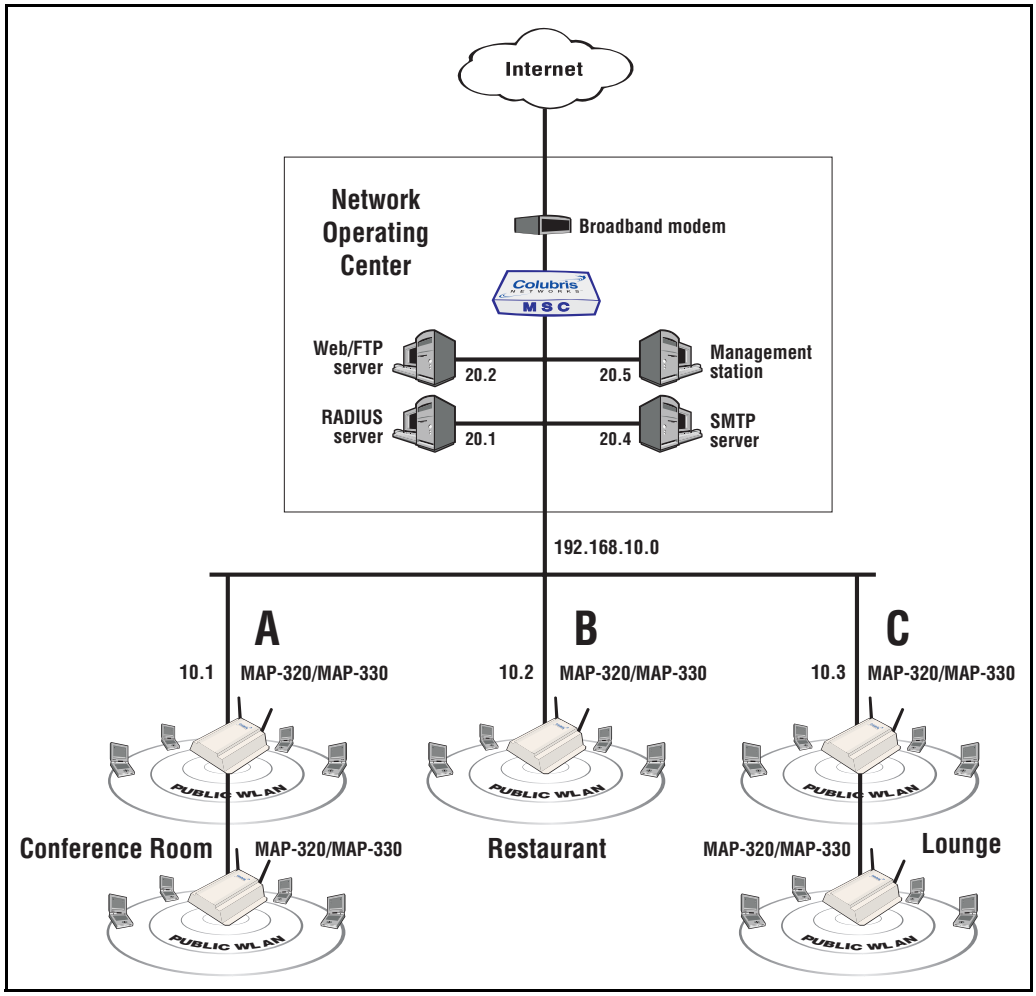
The SSID of the access point the customer is associated with (wireless only).

Colubris-specific attribute: GROUP

The GROUP of the access point the customer is associated with (wireless only).

Example

Consider the following topology for a fictional small hotel. The restaurant and lounge are available to all hotel customers who subscribe to the wireless service. However, the conference room is available only to a specific group of guests who book it in advance.



In this example, the access points in each area are assigned the following unique group names:

- conference_room
- restaurant
- lounge

When a customer logs in, server-side code can be used to determine the access point they are associated with by inspecting the Called-Station-ID. Then, using customer's account information, access can either be granted or denied.

Security

The MSC accepts location-aware information only from Colubris Networks satellites that have a matching shared secret to its own. Customers on other access points (Colubris or third-party) are treated as "wired".

iPass support

The MSC provides support for the Generic Interface Specification from iPass which enables you to create an iPass-compatible hotspot.

To set up the MSC as an iPass hotspot, you must define the IPass authentication server on the **Security > RADIUS** page.

Note: *The RADIUS Reply-Message can be retrieved when using NOC authentication by using the %r placeholder in login-err-url and welcome-url, and extracting it from the answer sent by the MSC upon a NOC authentication request.*

IPass login URL

This attribute has been replaced by the WISPr login URL attribute. It is still supported for backward compatibility. However, new development should use the WISPr login URL attribute.

This attribute lets you define the location of the IPass login page. The MSC automatically redirects customers with IPass client software to this page.

Use the following Colubris-AVPair value string:

`ipass-login-url=`*`URL_of_page`*

Where:

Parameter	Description
`URL_of_page`	Address of the IPass login page.

ASP functions

The following ASP functions can be called from the internal pages only.

Errors

GetAuthenticationErrorMessage()

Returns a message (from message.txt) indicating the status of the last authentication request. This function is used on the default Login and Fail pages to update the customer on the status of the login or logout.

RADIUS

GetMsChapV2Failed()

Returns the MS CHAP V2 error string. This function is only supported if you select MSCHAP V2 as the authentication scheme on the MSC (**Security > RADIUS** page). The RADIUS server must also support this feature. For a list of possible return values see RFC 2759.

GetRadiusNasId()

Returns the NAS ID configured for RADIUS Profile on the MSC. (See the Administrator's Guide for details on setting the NAS ID.) This can be used to identify the MSC that authenticated a customer. For an example of how this function is used, see GetNasAddress().

GetRadiusReplyMessage()

Returns the string sent by the RADIUS server when an authentication request fails. The RADIUS server must be configured to support this feature. The information contained in the returned string depends on the configuration of the RADIUS server.

GetNasAddress()

Returns the fully-qualified domain name of the MSC as is specified in the currently loaded SSL certificate.

Example

In certain instances you may want customers to register for an account before they log in. To accomplish this you could modify the Login page by adding a register button. This redirects the customer's browser to a registration web server where they can set up their account. (This page must be made accessible to non-authenticated customers using the appropriate access list rule.)

To avoid having the customer login once registration is complete, the registration web server can send the customer back to the MSC using a special URL that automatically logs the customer into the public access interface.

Assuming the registration server is 192.169.30.1, the register button code on the Login page might look something like this:

```
<FORM><INPUT
onclick="javascript:window.location='https://192.168.30.1/demo-php/
register.php?
NASip=<%GetNasAddress();%>&NASid=<%GetRadiusNasId();%>';"
type=button value="Click Here to Register">
</FORM>
```

The NAS ID and NAS address are required when the customer is redirected back to the MSC after registration. The code on the registration web page would look something like this:

```
// Registering user information in the backend database
RegisterUser($username,
$firstname,
$lastname,
$company,
$title,
$phone,
$email,
$NASid,   // identifies the MSC the customer is connected to
$NASip
);

// set URL to redirect browser to
$targetURL = "location: https://
" . $NASip . ":8090/goform/HtmlLoginRequest?
username=" . $username . "&password=" . $password;

// When done
header($targetURL);
```

The target URL is built using the NAS IP and username and password. The form name is hard-coded.

Page URLs

GetSessionUrl()

Returns the URL of the Session page.

GetWelcomeUrl()

Returns the URL of the Welcome page.

GetFailRetryUrl()

Returns the URL of the next internal page to display as follows:

- Returns the Fail page URL if a login or logout request is currently pending.
- Returns the Transport page URL if the customer is already logged in.

This function is designed to be used in conjunction with IsRequestPending().

GetOriginalUrl()

Returns the URL the customer tried to access before being redirected to the Login page.

Session status and properties

IsRequestPending()

Returns ‘yes’ if a login or logout request is already pending for the current customer. This function is useful when a RADIUS server is slow to respond and a customer repeatedly clicks the login or logout buttons. For example, consider the following code which could be used to modify the Fail page to address this problem.

```
function loading()  //called when the fail page is first loaded
if ("<% IsLoggedIn(); %>" == "yes") //logout is pending, so refresh page
refresh();
else
{
     // customer is already logged out or a login is currently pending
     // (i.e., customer clicked login button twice
if ("<% IsRequestPending(); %>" == "yes")
setTimeout('refresh()',3000);
else //no login or logout is pending and customer is logged out
document.form1.close.value = "Close window"; //change button label
}
```

```
}

function refresh()   // refresh the Fail page
{document.location="<%GetFailRetryUrl();%>"; }
```

IsLoggedIn()

Returns "yes" if the customer is logged in. See IsRequestPending() for an example that shows how to use this function.

GetSessionStateMessage()

Returns a message (from message.txt) indicating the status of the customer session.

GetUserName()

Returns the username for the current customer.

GetMaxSessionTime()

Returns the total amount of connection time configured for the current customer session in minutes and seconds in the format: mm:ss.

GetMaxSessionTimeHMS()

Returns the total amount of connection time configured for the current customer session in hours, minutes and seconds in the format: hh:mm:ss.

ConvertMaxSessionTime(unit)

Returns the total amount of connection time configured for the current customer in the specified unit.

y	Years
d	Days
h	Hours
m	Minutes
s	Seconds

For example if the customer account is configured for 5000 seconds, then:

- ConvertSessionTime("y") returns 0, calculated as (5000 / (365*24 *60*60)).
- ConvertSessionTime("d") returns 0, calculated as (5000 / (24*60*60)).
- ConvertSessionTime("h") returns 1, calculated as (5000 / (60*60)).
- ConvertSessionTime("m") returns 83, calculated as (5000 / 60).
- ConvertSessionTime("s") returns 5000, calculated as (5000 / 1).

TruncateMaxSessionTime(unit)

Returns the total amount of connection time configured for the current customer truncated to the specified unit.

y	Years
d	Days

h	Hours
m	Minutes
s	Seconds

For example if the customer account is configured for 5000 seconds, then:

- TruncateSessionTime("y") returns 0.
- TruncateSessionTime("d") returns 0.
- TruncateSessionTime("h") returns 1.
- TruncateSessionTime("m") returns 23.
- TruncateSessionTime("s") returns 20.

GetSessionRemainingIdleTime()

Returns the amount of time remaining until the customer is logged out due to inactivity.

GetSessionTime()

Returns session duration for the current customer in minutes and seconds in the format: mm:ss.

GetSessionTimeHMS()

Returns session duration for the current customer in hours, minutes and seconds in the format: hh:mm:ss.

ConvertSessionTime(unit)

Returns session duration for the current customer in the specified unit. See ConvertMaxSession time for details.

TruncateSessionTime(unit)

Returns session duration for the current customer truncated to the specified unit. See TruncateMaxSession time for details.

SetSessionRefreshInterval(sec)

Specifies the refresh interval for the Session page in seconds.

GetSessionRemainingTime()

Returns the amount of connection time remaining for the current customer session in minutes and seconds in the format: mm:ss.

GetSessionRemainingTimeHMS()

Returns the amount of connection time remaining for the current customer session in hours, minutes and seconds in the format: hh:mm:ss.

ConvertSessionRemainingTime(unit)

Returns the total amount of connection time remaining for the current customer in the specified unit. See ConvertMaxSession time for details.

TruncateSessionRemainingTime(unit)

Returns the total amount of connection time remaining for the current customer truncated to the specified unit. See TruncateMaxSession time for details.

GetMaxSessionIdleTime()

Returns the total amount of idle time configured for the current customer session.

GetSessionIdleTime()

Returns the amount of time the current session has been idle.

GetSessionInputPackets()

Returns the number of packets received by the current customer session.

GetSessionInputOctets(div)

Returns the number of octets received by the current customer session.

If you specify a value for the optional parameter div, then the return value is the number of octets divided by *div*.

GetSessionOutputPackets()

Returns the number of packets sent by the current customer session.

GetSessionOutputOctets(div)

Returns the number of octets sent by the current customer session.

If you specify a value for the optional parameter div, then the return value is the number of octets divided by *div*.

Session quotas

These functions let you retrieve the quota limits that are set for the current customer session. If any of these limits are reached, the customer is logged out. For details see "Quotas" on page 51.

GetSessionRemainingInputPackets()

Returns the number of incoming packets the current customer session can still receive. This value is a decimal string (10 characters) representing a 32-bit unsigned integer.

GetSessionRemainingInputOctets(div)

Returns the number of incoming octets the current customer session can still receive. This value is a decimal string (20 characters) representing a 64-bit unsigned integer.

If you specify a value for the optional parameter div, then the return value is the number of octets divided by *div*.

GetSessionRemainingOutputPackets()

Returns the maximum number of outgoing packets the current customer session can still send. This value is a decimal (10 characters) string representing a 32-bit unsigned integer.

GetSessionRemainingOutputOctets(div)

Returns the maximum number of outgoing octets the current customer session can still send. This value is a decimal string (20 characters) representing a 64-bit unsigned integer.

If you specify a value for the optional parameter div, then the return value is the number of octets divided by *div*.

GetMaxSessionInputPackets()

Returns the maximum number of incoming packets the current customer session can receive. This value is a decimal string (10 characters) representing a 32-bit unsigned integer.

GetMaxSessionInputOctets(div)

Returns the maximum number of incoming octets the current customer session can receive. This value is a decimal string (20 characters) representing a 64-bit unsigned integer.

If you specify a value for the optional parameter div, then the return value is the number of octets divided by *div*.

GetMaxSessionOutputPackets()

Returns the maximum number of outgoing packets the current customer session can send. This value is a decimal string (10 characters) representing a 32-bit unsigned integer.

GetMaxSessionOutputOctets(div)

Returns the maximum number of outgoing octets the current customer session can send. This value is a decimal (20 characters) string representing a 64-bit unsigned integer.

If you specify a value for the optional parameter div, then the return value is the number of octets divided by *div*.

iPass support

iPassGetLoginUrl()

Returns the iPass Login URL.

iPassGetAbortLoginUrl()

Returns the iPass Abort Login URL.

iPassGetLogoffUrl()

Returns the iPass Logout URL.

iPassGetRedirectResponseCode()

Checks if the iPass authentication server is reachable and enabled. Returns one of the following values:

0	Authentication server is reachable and enabled.
105	The authentication server could not be reached or is unavailable.

255	The authentication server could not be reached due to an error on the MSC (Internet port not up, for example).

iPassGetAccessProcedure()

Returns the access procedure supported by the MSC. The MSC supports procedure version 1.0.

iPassGetLocationName()

Returns the location name defined on the **Public access > Access control** page.

iPassGetAccessLocation()

Returns a value which can be used to determine the access point a customer is connected to. This is useful when you are using one or more MAPs in addition to the MSC.

- If a customer logs into a MAP, this function returns the MAC address of the MAP's downstream port.
- If a customer logs into the MSC, this function returns the MAC address of the MSC's LAN port.

iPassGetLoginResponseCode()

Returns one of the following values when a customer attempts to login to iPass:

50	Login was successful.
100	Login failed. Access was rejected.
102	Login failed. Authentication server error or timeout.
201	Authentication is pending.
255	The authentication server could not be reached due to an error on the MSC (Internet port not up, for example).

iPassGetLoginResponseCode()

Returns one of the following values when a customer attempts to logout from iPass:

150	Logout was successful.
255	The authentication server could not be reached due to an error on the MSC (Internet port not up, for example).