



Branding Guide

Release 3.10 and 4.1
July 2009
Part Number 010-00544f

This manual supports Messaging Operating System (MOS) release 3.10.3 and 4.1.3 and later MOS releases until replaced by a newer edition.

The Mirapoint Software and Mirapoint documentation are Copyright © 1998-2009 Mirapoint Software, Inc. All Rights Reserved. You may not print, copy, reproduce, modify, distribute or display this work in hard copy, electronic, or any other form, in whole or in part, by any electronic, mechanical, or other means, without the prior written consent of Mirapoint Software, Inc., except that you are permitted to make one copy for archival purposes only in connection with the lawful use and operation of this software.

Mirapoint, RazorGate, and the Mirapoint logo are registered trademarks of Mirapoint Software, Inc. Mirapoint Message Server, Mirapoint Directory Server, Mirapoint Operations Console, RazorSafe, DirectPath, WebMail Direct, WebCal Direct, and GroupCal Direct are trademarks of Mirapoint Software, Inc.

Mirapoint integrates third party software programs within the Mirapoint Software, which are subject to their own license terms. If the user does not agree to abide by the applicable license terms for the integrated third party software programs as defined by the Mirapoint Software License Agreement, then you may not install or operate the Mirapoint Software. These software license agreements, including the Mirapoint Software License Agreement, Red Hat End User License Agreement, and Open LDAP Public License, can be viewed at <http://www.mirapoint.com/licenses/thirdparty/eula.php>.

Portions of this product are Copyright © 1982, 1986, 1989, 1991, 1993 the Regents of the University of California. All Rights Reserved.

Portions of this product are Copyright © Dell Inc. Used with permission.

Portions of this product are Copyright © 2008 Red Hat, Inc. All Rights Reserved. The “Red Hat” trademark and the “Shadowman” logo are registered trademarks of Red Hat, Inc. in the U.S. and other countries.

Portions of this product are Copyright © 1997, 1998 FreeBSD, Inc. All Rights Reserved.

Portions of this product are Copyright © 1996-1998 Carnegie Mellon University. All Rights Reserved.

Portions of this product are Copyright © 1997-1998 the Apache Group. All Rights Reserved.

Portions of this product are Copyright © 1987-1997 Larry Wall. All Rights Reserved. See <http://www.perl.org>.

Portions of this product are Copyright © 1990, 1993-1997 Sleepycat Software. All Rights Reserved.

This software is derived in part from the SSLava™ Toolkit, which is Copyright © 1996-1998 by Phaos Technology Corporation. All Rights Reserved.

Portions of this product are Copyright © 1998, 1999, 2000 Bruce Verderaime. All Rights Reserved.

Portions of this product are Copyright © 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Macintosh is a trademark of Apple Computer, Inc.

Windows, Outlook, Exchange, and Active Directory are trademarks of Microsoft Corporation.

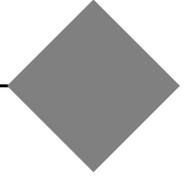
Java and Solaris are trademarks of Sun Microsystems, Inc.

Linux is a registered trademark of Linus Torvalds.

All other trademarks are the property of their respective owners.

OTHER THAN ANY EXPRESS LIMITED WARRANTIES THAT MIRAPOINT PROVIDES TO YOU IN WRITING, MIRAPOINT AND MIRAPOINT'S LICENSORS PROVIDE THE SOFTWARE TO YOU “AS IS” AND EXPRESSLY DISCLAIM ALL WARRANTIES AND/OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL MIRAPOINT'S LICENSORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY (INCLUDING NEGLIGENCE OR OTHER TORT), ARISING IN ANY WAY OUT OF YOUR USE OF THE SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF DAMAGES. MIRAPOINT'S LIABILITY SHALL BE AS LIMITED IN THE LICENSE AGREEMENT.

Contents



Preface	7
Assumptions	7
About Mirapoint Documentation	7
Getting Technical Support	7
Typographic Conventions	8
Iconic Conventions	9
1	
About Branding	11
System Requirements	11
Process Overview	12
Planning Your Brand/Creating a Brand Document.....	12
Setting Up Your Branding Environment.....	13
Determining Branding Method	13
Downloading and Saving the Factory Brand	15
Understanding the Factory Brand Files.....	16
Working with Branding Files and Directories	16
Working with the Cascading Style Sheets.....	18
Working with Variables.....	21
Working with Image Files	24
Customizing the Branding Files.....	24
Testing Your Brand.....	24
2	
Common Branding Tasks and Tips	25
Updating Brands	25
Three-Way Merge.....	25
Upgrading from Standard Edition to Corporate Edition	26
Creating a Domain Sensitive Login Page	27



Branding the Login Page Using JavaScript	27
Tips	29
About Quotes.....	29
Locale display_charset Values	30

3

Corporate Edition Branding 31

Locating and Branding the Style Sheets	31
Branding the Login Page	33
Corporate Edition Login Images	33
Corporate Edition Login Text	34
Branding the Banner.....	35
Branding Global Formats	36
Branding the Tabs	36
Branding the Links	37
Branding the Options Links	38
Branding Individual Fields of an Option Page	39
Branding the Calendar	40
Choose a Resource.....	40
Large Group Event Scheduling Limit.....	40
Locating and Branding the Help Files.....	41
Next Steps	42

4

Standard Edition Branding 43

Common Standard Edition Branding	44
Branding the Style Sheets.....	44
Branding the Login Page	44
Branding the Banner.....	45
Branding the Links Frame	46
Branding the Help Files.....	48
Branding the WebMail Direct Pages	48
WebMail Addressbook.....	49
Planning Your Custom Brand	49
WebMail Viewing Modes.....	50
About the WebMail Screen Elements	53
Frame Files Reference.....	54
How the WebMail Pages are Defined.....	59
More WebMail Branding	67
Setting a Default Charset for WebMail	67

Branding Default Mode Preferences for WebMail.....	67
Changing the Date Format in WebMail.....	68
Changing How WebMail Handles Attachments	70
Branding the WebCal Direct Pages.....	71
About the WebCal Pages	71
WebCal Addressbook	71
About the WebCal Viewing Modes.....	71
About the WebCal Screen Elements	73
How the WebCal Pages are Defined	73
More WebCal Branding.....	77
Supporting Group Calendar.....	77
Branding Default Mode Preferences for WebCal.....	77
Branding Calendar Email Messages	77
Branding the Options/Account Pages	81
About the Options/Account Pages	81
Options/Account Process Files	81
Options/Account HTML Template Files.....	82
Branding the Spelling Pages.....	83
About the Spelling Pages.....	83
Spelling Process Files.....	83
Spelling HTML Files.....	83
Branding the Address Book Pages	84
About the Address Book Pages	84
Address Book—WebMail vs. WebCal.....	84
Address Book Process Files	85
Address Book HTML Template Files.....	85
More Address Book Branding	87
Localized Import/Export in Address Book	87
Branding Contact Data	88
Adding LDAP Search Fields	90
Next Steps.....	91

5 Junk Mail Manager Branding 93

Common JMM Branding.....	93
Branding the Style Sheets	93
Branding the Login Page	94
Branding the Banner	95
Branding the Links Frame.....	96
Branding the Help Files.....	97
Branding the Junk Mail Manager Pages	98
About the Junk Mail Manager Pages	98
Spam Frameset Definition Files.....	98



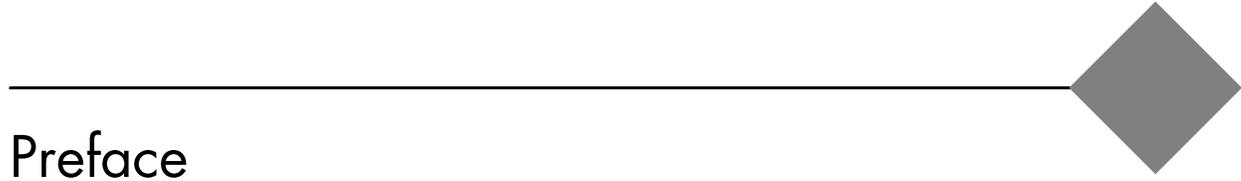
Spam Process Files.....	99
Spam HTML Files.....	99
Next Steps.....	101

6

Publishing a Brand	103
Viewing Your Brand	103
Zipping Your Brand Files.....	103
Publishing Your Brand Files	104
Assigning Your Brand	104

7

Troubleshooting.....	105
Error Messages.....	105
URLs Versus Forms.....	105
Duplicate Variable Keys	105
Publishing Problem	105
Failing to Display the Administration Suite Login	106
Restoring the Factory Brand.....	106
From the Administration Suite	106
From the CLI	106
Index	107



Preface

Welcome to the Mirapoint *Branding Guide* for release 3.10 and 4.1. This book is intended for administrators of a Mirapoint messaging system.

Assumptions

This book assumes you have a good understanding of HTML, CSS, JavaScript, and Mirapoint messaging. If you are not very familiar with HTML, it is advisable to spend at least an hour or so reading through the HTML primers available at

<http://www.w3c.org>

About Mirapoint Documentation

Documentation for all Mirapoint products is available through the Information Library on the Mirapoint Support website:

<https://support.mirapoint.com/>

The Information Library provides the hardware and software documentation for all supported Mirapoint releases and appliances, and the Support Knowledge Base. The Support site is accessible to all customers with a valid Support Contract. If your company has a valid contract but you need a Support login ID, email support@mirapoint.com.

Getting Technical Support

If you experience problems with your appliance, contact the company from which you purchased your Mirapoint appliance.

If you purchased your appliance directly from Mirapoint, contact Mirapoint Technical Support by email, telephone, or via the Mirapoint Support website:

Email: support@mirapoint.com
(China) support@mirapoint.com.cn

Telephone:

- ❖ (USA) 1-877-MIRAPOINT (1-877-647-2764)
- ❖ (Outside the USA) +1 408-720-3800
- ❖ (UK) +44 (0) 1628-535699
- ❖ (China) +400 707-1086

❖ (Australia) +1 800-633784

Website: <https://support.mirapoint.com/>

When contacting Technical Support, be prepared with the following information about your appliance:

Table 1 Appliance Information for Technical Support

	MOS CLI command (Message Server, RazorGate)	AOS UI Location (RazorSafe)
Software release	Version	In the Status tab, select System Info .
Host ID	License Hostid	In the Status tab, select System Info .
Serial number	Model Get Serial	In the Status tab, select System Info .
Hardware model	Model Get Chassis	In the Status tab, select System Info .

Typographic Conventions

[Table 2](#) describes what the different fonts and typefaces indicate in this manual.

Table 2 Typographic Conventions in This Manual

	Use	Example
Bold	User interface elements	From the File menu, select Save As...
<i>Italic</i>	Definitions, emphasis, or titles	<i>A folder</i> is a container that stores email messages. Specify <i>at least two</i> DNS servers. See the <i>Message Server Administrator's Guide</i> .
Courier	Screen display text, command names, or text to type *	Enter your IP address: Use the License Hostid command. At the prompt, type Version.
<i>Courier Italic</i>	Variables for which you substitute when you type	<i>your_IP_address</i>

* Command-line interface (CLI) commands are case-insensitive. For readability, commands in this manual are shown in mixed case (for example, License Hostid).

Iconic Conventions

Table 3 describes what the different icons in this manual indicate.

Table 3 Iconic Conventions in This Manual

	Use
	Important information
	Critical information
	Note information that <i>should</i> be read
	License required
	Potential of causing bodily harm (hardware only)



About Branding

Branding is the process of modifying the look of an application by editing style sheets, images, etc. Branding can be applied system-wide to the primary domain, and applied individually to delegated domains.

This manual provides branding information and various procedures for the following Mirapoint applications:

- ◆ WebMail/WebCal Corporate Edition
- ◆ WebMail/WebCal Standard Edition
- ◆ Junk Mail Manager (JMM)

This chapter provides information on the following high-level topics:

- ◆ [System Requirements](#)
- ◆ [Process Overview](#) on page 12
- ◆ [Planning Your Brand/Creating a Brand Document](#) on page 12
- ◆ [Setting Up Your Branding Environment](#) on page 13
- ◆ [Understanding the Factory Brand Files](#) on page 16
- ◆ [Customizing the Branding Files](#) on page 24
- ◆ [Testing Your Brand](#) on page 24

System Requirements

The branding process requires the following systems configurations:

- ◆ Mirapoint appliance configuration
 - ❖ Hostname and HTTP access to the Mirapoint appliance
 - ❖ System administrator login and password
 - ❖ Licenses installed for the user applications to be branded (WebMail or WebCal)

To see the user application pages, go to:

`http://hostname/wm` for WebMail SE (WebCal SE also, if licensed),

`http://hostname/mc` for WebCal SE,

`http://hostname/wm/eml/login.html` for WebMail/WebCal CE

`http://hostname/spam` for Junk Mail Manager

For the Administration Suite account management for users, go to:

`http://hostname/acctadmin`

- ◆ Windows or UNIX workstation configuration:
 - ❖ Zip file utility
 - ❖ Simple HTML text editor



You can use any HTML editor as long as it does not add any extraneous information or change the format of the files, which can result in errors when the files are published.

- ❖ System Brand file set ([Downloading and Saving the Factory Brand](#) on page 15)

Refer to [Setting Up Your Branding Environment](#) on page 13 for details.

Process Overview

The branding process involves customizing the downloaded factory files followed by the publishing of the customized files. The factory files are downloaded as a zip file that includes the factory application file sets of HTML templates, process files, images and online help files.

The branding process involves the following tasks:

1. Planning Your Brand/Creating a Custom Brand Document
2. Setting Up Your Branding Environment
3. Customizing the Branding Files
4. Testing Your Brand
5. Zipping Your Brand Files
6. Publishing Your Brand Files
7. Assigning Your Brand

Planning Your Brand/Creating a Brand Document

As you plan your brand, create a Custom Brand Document. The Custom Brand Document is your record of the changes implemented to produce your custom brand. This document will be very helpful in the future when you upgrade your MOS and possibly need to re-brand.

For each screen element that you are going to brand do the following:

- ◆ Determine what color scheme you want for your brand.
- ◆ Gather together image files that you want to use.
- ◆ Determine what text changes need to be made and what text you want to use.
- ◆ List the files, variables, HTML elements, and so forth that you are going to change.
- ◆ Detail how you are going to make the changes.

For example, if you are going to change the background colors used in the links and banner files; describe those changes along with what files you changed and what new colors you used.

- ◆ Document the MOS version the brand was created from.
- ◆ Name your brand document something like *branding_notes.txt* and save it with the modified brand in the `extras` directory.

Setting Up Your Branding Environment

Setting up your branding environment involves the following tasks:

1. Ideally on another system, make a secure copy of your system application file set.
2. Chose a branding method.
 - ❖ System Brand—Set up for one system brand
 - ❖ Named Brand—Set up for multiple delegated domains brands
3. Download the factory brand.
4. Unzip the factory brand file set.
5. Copy a brand file set for each brand you will make.

Determining Branding Method

There are two methods of publishing a brand:

- ◆ **System Brand:** one domain automatically uses the “system” brand that you publish unless otherwise specified.
- ◆ **Named Brand:** use a file set identical to the system brand for multiple domains, which provides the ability to use different **Login** pages per domain; and provides the ability to set the over quota message per brand or per domain.

Depending on how many brands you plan to do at one time, follow either the [System Brand Setup](#) or [Named Brand Setup](#) procedures.

System Brand Setup

If you plan to work on one brand at a time, there is no additional setup. System brand setup uses one brand published and viewed as a System brand. If this is your plan, skip to [Downloading and Saving the Factory Brand](#) on page 15.

Named Brand Setup

Named brands allow you to create multiple brands, and assign domains to them. Named brands are viewed by logging into different domains using the “at” (@) symbol and domain name in the **Username** login box.

Use the named brand setup if:

- ◆ If you plan to work on more than one brand at a time, or
- ◆ If more than one person is working on branding at the same time.

The named brand setup involves the following tasks:

1. On the Mirapoint appliance, create the multiple delegated domains ([Creating a Delegated Domain](#) on page 14)
2. Configure the local hosts ([Configuring Your Local Hosts File](#) on page 15)
 - ❖ on your workstation to access the domains, or
 - ❖ on a DNS server as you would in a production environment

[Table 4](#) describes a possible configuration scheme that you could use to set up multiple delegated domains and the procedure follows.

Table 4 Example Multiple Brand Environment

Domain Name	Administrator	Login Password	Login User name
domain1.com	admin	mypass1	admin@domain1.com
domain2.com	admin	mypass2	admin@domain2.com
domain3.com	admin	mypass3	admin@domain3.com

Creating a Delegated Domain

To create a delegated domain:

1. As an administrator logged in to the Administration Suite, go to **Domains > Administration**.
2. On the **Administer Domains** screen, provide the following details:
 - ❖ **Domain Name:** The name of a fully qualified domain. A fully qualified domain name includes the host name as well as the domain name (hostname.domain-name.com).
 - ❖ *(Optional)* **Domain Disk Quota:** The amount of available domain-wide disk space (in KB) for the delegated domain.
 - ❖ *(Optional)* **Maximum Users:** The maximum number of domain users that can be added to this domain. **Note:** Each user account on a domain requires space allocation for mail and puts a load on the network when actively in use. To prevent overtaxing a system and network, a delegated domain can be assigned a user limit. It is also possible to set a disk quota for a delegated domain to maximize performance.
3. Click **Add Domain**.

The name of the new delegated domain appears in the Domain list.

Configuring Your Local Hosts File

By configuring your workstation host file, you are mapping the IP address of the Mirapoint Sever to the domain names you created earlier, which enables you to view multiple named brands on one server without setting up a DNS server.



Configure the hosts file on every workstation that the domain brand is viewed.

To configure your local hosts file:

1. Open the local hosts system file.
 - Windows**—open C:\WINNT\system32\drivers\etc\hosts
 - UNIX**—open the comparable file; for example, '/etc/hosts'
2. Enter the IP address of the Mirapoint appliance.
3. Enter the domains you created in the format specified by the file.
4. Save the file.

Downloading and Saving the Factory Brand

The factory installed brand file set that you download uses the file naming convention `x.y_sysbrand.zip`, where `x.y` represents the installed version of the Messaging Operating System (MOS). Use the Administration Suite to download the factory brand zip file.

To download the factory brand zip file:

1. Log in to the Administration Suite as an administrator and go to **System > Branding > Download**.
2. Select the brand file appropriate for your system, DOS or UNIX. In order to reduce download and publishing time, select only the components that you want to brand.

You can choose from the following components:

- ❖ **Online Help:** These are the files for Corporate Edition and Standard Edition online Help. These files are fully brandable.
 - ❖ **Corporate Edition:** These are the files for WebMail Corporate Edition. These files have limited brandability.
 - ❖ **Standard Edition (Includes Junk Mail Manager):** These are the files for WebMail Standard Edition and Junk Mail Manager. These files are fully brandable.
3. Click **Download**.
 4. Save and un-zip the file.

Understanding the Factory Brand Files

This section describes the files and directory structure contained within the `sysbrand.zip` file that you downloaded and saved to your workstation and explains how to work with them generally.

Working with Branding Files and Directories

The file directory structure of the system brand zip file is important to maintain.

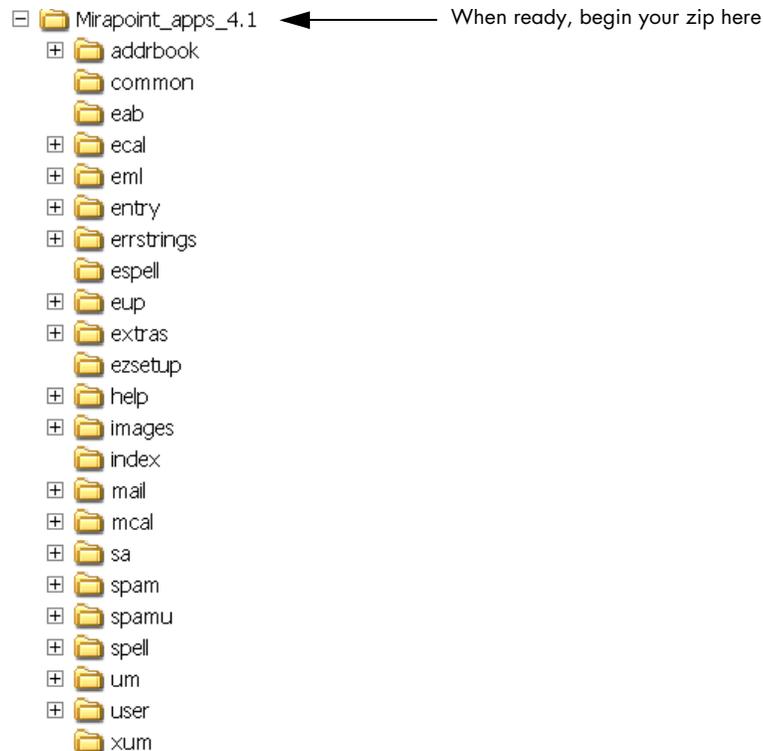


Figure 1 System Brand Zip Directory Structure

Each web page you see, with the exception of the online help, is dynamically generated and composed of HTML files that use CSS (cascading style sheets) and **var**tab (variable definition) files, and at least one string dictionary file called **locale.var**. Other files, for example `toc.js`, provide functionality. In this document the HTML files are referred to as **templates**; the other files are referred to as **process files**. The branding files you work with include:

- ◆ Standard Edition & Junk Mail Manager CSS files in `extras/css`:
 - ❖ **jmm_style.css**: Used by Junk Mail Manager as a cascading style sheet for `style.css`. In the system brand zip these two style sheets are identical; make changes to `jmm_style.css` to change display elements in Junk Mail Manager.
 - ❖ **style.css**: Used by WebMail/WebCal Standard Edition pages.

- ◆ Corporate Edition CSS files in **extras/css**:
 - ❖ **ie.css**: Used as a cascading style sheet for **uui.css** (Corporate Edition specific style sheet) for special Internet Explorer issues.
 - ❖ **mozilla.css**: Used as a cascading style sheet for **uui.css** (Corporate Edition specific style sheet) for special Mozilla browser issues.
 - ❖ **uui.css**: Used by WebMail/WebCal Corporate Edition pages.
 - ❖ **system-*.css** (located in the **themes/** sub-directory of the **extras/css** directory): Used to create the different color themes available in WebMail/WebCal Corporate Edition.



The Administration Suite pages use the following unbrandable style sheets: **admin.css**, **rg_style.css**, **rgez_style.css**, **sa45_admin.css**, **styleMpHelp.css**, **styleMpHelpHome.css**, **styleRgHelp.css**, and **style.RgHelpHome.css**.

- ◆ Vartab files: There are several types of vartab files: **cell.var**, and **locale.var**, and application-specific vartab files, located in each application **en_US** sub-directory, including **errstrings.var**, located in the **locale** sub-directory of the **errstrings** directory; these can only be branded for WebMail/WebCal Standard Edition or Junk Mail Manager. Only the **brandvars.var** file located in the **extras** directory can be modified for Corporate Edition Branding WebMail/WebCal.
- ◆ Image files: Located either in the top-level **images** directory, **images/eui** directory (Corporate Edition only), or in a **locale/locale name/images** sub-directory in an application directory.
- ◆ Help files: Located in the **help/locale/locale name/** directory, used as follows:
 - ❖ WebMail/WebCal Standard Edition: Online Help files are in the **addrBookCom**, **calendar**, **images**, **user**, and **webmail** sub-directories.
 - ❖ WebMail/WebCal Corporate Edition: Online Help files are in the **eml_I_Help/Output** and **eml_W_Help/Output** sub-directories (“I” for the integrated WebMail/WebCal Help files; “W” for the WebMail only version)
 - ❖ Junk Mail Manager: Online Help files are in the **spam**, and **images** sub-directories.
- ◆ HTML template files: Located in each application **locale name** sub-directory. At a minimum, the following file directories are required:
 - ❖ WebMail/WebCal Standard Edition:
 - **addrbook/**
 - **entry/**
 - **errstrings/**
 - **extras/**
 - **images/**
 - **mail/**
 - **mcal/**
 - **spell/**
 - **user/**
 - ❖ Junk Mail Manager:
 - **extras/**
 - **images/**

- spam/
- spamu/
- ❖ WebMail/WebCal Corporate Edition:
 - common/
 - eab/
 - ecal/
 - eml/
 - errstrings/
 - espell/
 - eup/
 - extras/
 - images/



If you are not branding the help files or any non-English locale, you can delete the untouched **help** directory and the untouched **locale** directory, for example: ja_JP.utf-8. This will save uploading time. Once published, the default factory locale and help files automatically get used for those deleted directories.

Working with the Cascading Style Sheets

The usage of CSS (cascading style sheets) allows formatting to be handled in a centralized manner. The style sheets only define body, links, button, table header and table data row, background and text colors and styles. The factory brand zip file includes the following CSS files, in the **extras/css** directory:

- ◆ **style.css**, in the **extras/css** directory: WebMail/WebCal Standard Edition style sheet; also used by Junk Mail Manager and the WebMail/WebCal Standard Edition online Help files.
- ◆ **uui.css**, and **system-*.css** in the **extras/css** and **extras/css/themes** directories: WebMail/WebCal Corporate Edition style sheets.
- ◆ **ie.css** and **mozilla.css**, in the **extras/css** directory: Used as a cascading style sheet for uui.css (Corporate Edition specific style sheet) for special Internet Explorer or Mozilla browser issues, respectively.
- ◆ **jmm_style.css**, in the **extras/css** directory: Junk Mail Manager cascading style sheet (cascades off **style.css**).
- ◆ **document.css** and **catalog.css**, in the **help/eml_I_Help/Output** and **eml_W_Help/Output** directories: Corporate Edition WebMail/WebCal (eml_I_Help) and Corporate Edition WebMail only (eml_W_Help) online Help style sheets.
- ◆ **admin.css**, **rg_style.css**, **sa45_admin.css**, **safari.css**, **styleMpHelp.css**, **styleMpHelpHome.css**, **styleRgHelp.css**, **styleRgHelpHome.css**: These are used by the Administration Suite for which branding is not supported. Ignore these files.

Also included in `extras/css` are these explanation files; open them in a browser to get a visual of the style definitions:

- ◆ `usage.html`: Standard Edition and Junk Mail Manager
- ◆ `ce_usage.html`: Corporate Edition

CSS1 is the bare minimal approach used for the factory system brand. Support for CSS2 is largely a function of the browser, and the imagination of the branding professional.

The CSS provides the ability for you to quickly brand the colors throughout the application. You can change the background color of the pages, table header and data row and alternate table row color, link colors, page title color, and so forth. You can use a gif reference instead of a color with a CSS reference like this:

```
<BODY BACKGROUND=image.gif>
```



Within the HEAD tags of each HTML template is the CSS reference, `$(res_head)`. This reference is defined in `cell.var` and directs the application pages to the `style.css` file. You can change the value to use a different CSS file in `cell.var` (Custom Branding only).

About CSS Terminology

The CSS defines selectors, or classes, with rules. Included below is a brief explanation of these terms. See <http://www.w3.org/TR/REC-CSS1> for more information on CSS.

- ◆ **rule**: Example:

```
BODY { color: #000000 }
```

A rule consists of two main parts: the **selector** (example: 'BODY') and the **declaration** (example: 'color: #000000'). The selector identifies what elements the corresponding rule applies to; it can be a simple selector (example: 'BODY') or a contextual selector (example: 'BODY A') that consists of several simple selectors. The selector is the link between the HTML document and the style sheet, and all HTML element types are possible selectors. The declaration has two parts: **property** (example: 'color') and **value** (example: '#000000'). CSS1 allows many different property values to be defined.

- ◆ **class**: HTML attribute for defining child elements. For example, all elements inside the 'BODY' element can be classed, and the class can be addressed in the style sheet:

```
<HTML>
<HEAD>
  <TITLE>Title</TITLE>
  <STYLE TYPE="text/css">
    H1.pastoral { color: #00FF00 }
  </STYLE>
</HEAD>
<BODY>
  <H1 CLASS=pastoral>Way too green</H1>
</BODY>
</HTML>
```

The normal inheritance rules apply to classed elements; they inherit values from their parent in the document structure. One can address all elements of the same class by omitting the tag name in the selector:

```
.pastoral { color: green } /* all elements with CLASS pastoral */
```



It is important to remember that changes you make using the Cascading Style Sheets (CSS) functionality can have different results depending on what browser is used to view the finished files.



In the CSS, background colors are called “background” and text colors are called “color.”

About usage.html and ce_usage.html

While the selectors defined in style.css and uui.css are clearly described, the usage.html file (Standard Edition) and ce_usage.html file (Corporate Edition) are helpful for viewing your CSS branding changes.

usage.html—references style.css

To access usage.html (Standard Edition), go to:

```
extras/css/usage.html
```

ce_usage.html—references uui.css and themes/system-default.css

To access ce_usage.html (Corporate Edition), go to:

```
extras/css/ce_usage.html
```

About the CSS Fallback: brandvars.var

The CSS Fallback functionality ensures that the style sheet definitions are implemented in browsers that do not support style sheets. Some older web browsers do not support CSS, or support CSS disabling.

A special vartab file called “brandvars.var” ensures that your branded web pages display correctly without CSS. There is a corresponding “miravar_” variable defined in the brandvars.var file, known as the “CSS Fallback,” for each class defined in the style sheet. This brandvars.var file is used as the Corporate Edition Branding Fallback and the Standard Edition Branding Fallback as follows:

- ◆ For each instance of the `class=class` markup, a corresponding variable is declared in the brandable template file.
- ◆ The factory template files use brandvars.var that enable you to define your own variables.
- ◆ The substitutions declared are performed during the publishing of the brand to eliminate human error.

Table 5 Example of the CSS Fallback for the genHeader Class

In this file...	It's used like this...
HTML template WITHOUT fallback implemented	<code><td class='genHeader'\$(miravar_genHeader)></code>

Table 5 Example of the CSS Fallback for the genHeader Class

In this file...	It's used like this...
HTML template WITH fallback implemented	<code><td class='genHeader' color='#000000' bgcolor='#b0c4be'></code>
brandvars.var file	<code>"miravar_genHeader" = "color='#000000' bgcolor='#b0c4be'";</code>

Having different color values in the CSS vs. the brandvars.var file will give different results; make synchronous changes to the CSS and the corresponding miravars_ values.

You can also use the brandvars.var file to create your own variable definitions using the `custvar_` syntax. Keys to be substituted must begin with `"custvar_"`. For example:

```
"custvar_hostname" = "www.example.com";
```

Sample usage in an HTML template would be:

```
...<a href="${custvar_hostname}">...
```



Never re-use a previously branded brandvars.var file associated with a different MOS version. Always use the brandvars.var file associated with the MOS version that you are branding.



There is a bug in Netscape 4.X that is exposed if the above-mentioned recommendation is not followed. In Netscape 4.X, the 'color' attribute in the CSS is not recognized when specified in the BODY tag. However, the 'color' attribute is recognized in all of the other defined CSS selectors in Netscape 4.X. If you keep the CSS and brandvars.var definitions in sync you will not trigger this Netscape 4 bug.

Working with Variables

This section describes the different variables and vartab files and how they are used; this information mainly applies to Custom Branding. The only vartab file you can change with Corporate Edition Branding is the brandvars.var file.

About Variables

Many text elements on the web pages are presented through variables that are defined in the vartab files.

There are two types of variables available for branding used by the application files (detailed explanations follow):

- ◆ Publish-time variables: Are not effected by configuration schemes.
- ◆ Run-time variables: Are effected by configuration schemes.

Publish-time Variables

Publish-time variables are so called because their values are instantiated when the brand is published. **Publishing** is the act of making available over the web the branded application files. All publish-time variables are defined in the brandvars.var file.

Publish-time variables are more flexible than run-time variables because they do not depend on any configuration schemes. With publish-time variables you can:

- ◆ Change the value of any one.
- ◆ Create new ones of your own.

There are two default types of publish-time variables: **miravar** and **custvar**; both are defined in the `brandvars.var` file in the `extras` directory. Miravar variables ship with the application files as a fallback to the CSS definitions so that pages rendered in a non-CSS enabled environment retain the brand look. Also, the `brandvars.var` file contains the only variables that are available for Corporate Edition Branding.

Rendering is how the browser reads the code it receives over the Internet to make the pages display. The ability to define `custvar` variables is included as a convenience. No such variables are defined by default, but you can add them to your application files if you want.

Run-time Variables

Run-time variables are so called because their values are instantiated at run-time. **Run-time** is when the system sends the requested web page out over the Internet.

Run-time variables often depend on configuration schemes and are largely defined by the system. Many run-time variables define entire screen elements such as tables or table rows of data. Often a screen element, such as a table, is defined partly in the template; for example, the names of the column headers; and partly with variables defined in `cell.var` that supply the data the table contains. With run-time variables you can:

- ◆ Change the value of some; many are system-used variables that you cannot change. Also, you *cannot* create any run-time variables.
- ◆ Move them around on the page.

Publish-time, Run-time, and Render-time

When a brand is *published* the system parses every file with these extensions: `.html`, `.htm`, `.wml`, `.var`, `.js`, `.css` for `miravar_` or `custvar_` variables and replaces those variables with the values it finds in the `brandvars.var` file.

When a brand is *run*, the system replaces any remaining variables with the values it finds in the appropriate `vartab` file using the appropriate key; which key goes with which variable is determined by the Mirapoint system and by factors such as licensing. The variables sometimes, but not always, use the same or similar wording as the corresponding key.

When a brand is *rendered* in a style sheet-enabled browser, the browser reads the styles defined in `style.css` and implements those styles. When a brand is rendered in a no-style-sheet browser, the browser simply reads the files; the definitions are already in place because the system put them there at publish time. Once a page is rendered it contains no variables, they have all been filled in with a value.

About System Variables

Throughout this document the term “system variable” is used to refer to variables that are used by the system and are not available for branding. Many system variables are defined by the system at login, such as `$(uid)`, `$(sessionid)`, or `$(dflltime)` (user ID, session ID, and default time; respectively), and are used by the system in many places. Some system variables generate rows of table data or populate pages with data where needed. Other system variables are defined by user input, are saved by the system, and trigger system actions. For example, `$(read_sel)` tells the system that the user has chosen to mark a selected message with the **Read** flag.



You can move system variables around on the page and you can sometimes define their user-visible text or format, but you cannot delete system variables; doing so will cause malfunctions. Additionally, system variables are often page specific; that is, a system variable that works on one page will not work on another page. In general, system variables are not defined in `cell.var`; if you come across a variable and cannot find a definition for it, it is probably a system variable and should be left alone.

Variable Format

All variables use this format:

```
$(variable)
```

Variables may be defined only in a `vartab` file or by the system; for example, `$(user)`. The publish-time variables, `miravar` and `custvar`, must be defined in `brandvars.var`. All variable definitions are in the form of **key:value pairs**. Key:value pairs follow this format:

```
key = "value";
```

For example:

```
res_head = "<LINK REL=STYLESHEET TYPE='text/css' HREF='/extras/css/style.css'>";
```

The **key** being on the left of the equal sign (=) and the **value** being on the right and ending with a semicolon. The value is generally enclosed in double quotes to allow for spaces between words. The value may be a single word, a phrase, a table row, or an entire table, and may include other variables. If either the key or value contains a character that could be misinterpreted, put double quotes around the entire phrase.



All quotes inside of double quotes must either be single quotes, or they must be escaped with a backslash (\). See [About Quotes](#) on page 29 for more details.



Only the variables that appear in a factory HTML file are guaranteed to work, and then only on that page. A variable defined in `cell.var` will not necessarily work on any other page than the one for which it was designed.

Common Resource Names

There are two common resource names used throughout the application files, a HEAD tag: `res_head` (CSS link), illustrated as follows; and the `miravar_` variables (`brand.vars`) family.

Key:value pair for **res_head** in application cell.var file:

```
res_head = "<LINK REL=STYLESHEET TYPE='text/css' HREF='/extras/css/style.css'> ";
```

Sample usage in pre-published application HTML file:

```
$(res_head)
```

Published application HTML file, **res_head** variable expands to:

```
<LINK REL=STYLESHEET TYPE='text/css' HREF='/extras/css/style.css'>
```

Working with Image Files

The **images/**, **images/eui/**, and **app name/locale/locale name/images/** sub-directories for each application, contain the image files for that application. Typically in **gif** or **jpg** format. You can substitute your own images for use in your customized web pages. For domain branding, images specific to the brand can be placed in the **images** sub-directory inside the **locale** sub-directory, or can be placed in the top-level **images** directory. To reference images stored in the top-level location, you must include the following path within the image tag in the HTML page:

```
.
```

Powered by Mirapoint

On the Standard Edition **Login** page and the Corporate Edition **Login** page and banner, is the Mirapoint corporate copyright. When you brand those pages, the Mirapoint copyright should be replaced with a “Powered by Mirapoint” logo. Official Logo art is available at the Mirapoint website ([Figure 2](#)).

<http://www.mirapoint.com/news/logos.php>



Figure 2 Powered by Mirapoint Logo

Customizing the Branding Files

Customize the downloaded factory brand files following the procedures in the appropriate chapter. Depending on which applications you are branding, refer to either [Chapter 3, Corporate Edition Branding](#), or [Chapter 4, Standard Edition Branding](#).

Testing Your Brand

To test view your customized brand you must publish it. Refer to [Publishing a Brand](#) on page 103 for details.

Common Branding Tasks and Tips

This chapter describes some branding tasks common to the Corporate Edition and Standard Edition and Junk Mail Manager applications.

Updating Brands

When upgrading a Mirapoint application files' brand there are two approaches you can take:

- ◆ **Re-brand the new factory brand:** If your brand is fairly simple and/or if the factory brand has changed substantially, it may be easier to re-brand the new version; especially when you are using CSS. Simply update your system and copy over your CSS and any image files you are using and your brand upgrade is complete.
- ◆ **Incorporate into your brand the new version features and functions:** If your brand is very complicated it may be necessary to incorporate the new features and functions into your existing brand. This is the less preferable way as it involves much combing of the application files for the new changes and determining how to integrate them into the changes you have already made.



Never publish a brand based on a different MOS release. It can lead to functionality loss, broken links, and incorrectly displayed pages. For example, do not publish a MOS 3.8.7 brand onto a MOS 4.1.7 appliance.

Three-Way Merge

A “Three-Way Merge” may be required if your brand is significantly different from the factory brand and you want to incorporate into your brand new features and functions. A “Three Way Merge” involves the following procedures:

- ◆ Upgrade your system to the version that you want to brand. Download a factory brand zip file and diff it against the previous version factory brand zip file. For example, upgrade your system to 3.10, download the 3.10 factory brand and diff it against the 3.8 factory brand file set. You will then have a diff of the changes made to the factory brand.
- ◆ Now, diff your non-updated custom brand against the same version factory brand. In other words, if the custom brand that you want to update to 3.10 is a 3.8 brand, diff it against a 3.8 factory brand; you can get a copy of a previous release factory brand from Customer Support if you did not save one. You now have a diff of the changes you made to the factory brand.

- ◆ Compare the changes you made to the factory brand to create your custom brand with the changes that Mirapoint made to the factory brand to create the new release; where these changes overlap, you will need to update your brand.



Brands are not impacted by upgrading between dot-dots in a release, but Mirapoint discourages publishing older brands (from older major releases) to new major releases. Re-branding using the new major release's brand files will guarantee that the features are available for use.

Upgrading from Standard Edition to Corporate Edition

If you want to make WebMail/WebCal Corporate Edition your default mail service, follow these steps.

1. Upgrade your system to Mirapoint 3.8 or greater.
2. Enable Corporate Edition using the **System > Services > WebMail** page. Using the command line interface (CLI): **Conf Enable enterpriseui**
3. Set the login root to Corporate Edition using the **System > Services > HTTP > Main Configuration** page. Using the CLI: **Http Set Root enterpriseui**
4. If you currently have a brand, delete it; first select the brand on the **System > Branding > Select Brand** page, then delete it on the **System > Branding > Delete Brand** page. Using the CLI: **Brand Delete *brandname***
5. Download a system zip brand file using the **System > Branding > Download** page. Un-zip the file and make your changes. Using the CLI: **Brand Get *brandname URL***
6. Once you've made your changes, zip the files—maintaining the original directory and file structure—and publish the brand using the **System > Branding > Publish** page as either your “system” brand or a “named” brand, see [Determining Branding Method](#) on page 13 and [Testing Your Brand](#) on page 24 for details. Using the CLI: **Brand Add *brandname URL***
7. Lastly, assign domains to any named brand (all unassigned domains use the system brand by default) using the **System > Branding > Assign Brand** page. See the online Help for details on using the **System > Branding** pages. See also [Creating a Domain Sensitive Login Page](#) on page 27 for related information. Using the CLI: **Brand Assign *domain brandname***

Creating a Domain Sensitive Login Page

You may have all domain users share a **Login** page and view their domain's brand only after log in. Or, you may create a **Login** page for each brand and, when a domain that is assigned to that brand is accessed through the domain's own URL, the branded **Login** page displays. Each brand may have its own **Login** page, accessible by a URL including the domain name. The following rules apply:

1. The domain must be assigned to a brand.
2. The domain must be entered in DNS as a canonical name for the **A** record of the host.
3. Sub-domains inherit the brand assignment of the parent domain. For example, if `abc.com` is assigned to **brand_a**, then `xyz.abc.com` also uses **brand_a** without needing to be explicitly assigned to it. If you wanted `xyz.abc.com` to have a different brand, then you would need to use the CLI and assign it to that brand.
4. Text matches are used, not IP addresses.

The domain **Login** page uses the text from the URL to determine the brand. The user of a delegated domain must log in using their login and domain name (**user@domain.com**), as usual. Three applications have **Login** pages: WebMail, WebCal, and Options/Account.

Use the Administration Suite **System > Branding > Assign Brand** page to assign domains to brands. A domain brand assignment has the following properties:

- ◆ When a user logs in as **user@domain_name.com**, they see **brand_name's** brand.
- ◆ If there is a DNS entry for **domain_name.com** which points to the Mirapoint MAS, then the user sees **brand_name's** branded **Login** page when they access the domain's URL/**Login** page (including delegated domains).
- ◆ The domain administrator can see **brand_name's** over-quota message in the **Domain** pages of the Administration Suite; and can modify it, at which point they get their own copy. If they revert, they see **brand_name's** over-quota message again. There is no way for a domain administrator to see the system over-quota message unless they are assigned to the system brand.

Branding the Login Page Using JavaScript

You can have a single Message server that presents different **Login** pages depending on the hostname you reached it by, and have different users have different WebMail branding without making them type in their full email address to log in. You create this using JavaScript; follow these steps (please see the end of this section for an important note).

1. Make `login.html` redirect to different pages depending on the hostname. Add this JavaScript code to the top of `mail/.../login.html`:

```
<script type="text/javascript" language="javascript"> location =
window.location; hostname = location.hostname; location.pathname = "/"
extras/" + hostname + "_std.html"; </script>
```

This JavaScript takes the hostname from the URL that the user typed in and puts it in the path part of the URL—think of it as client-side redirect. So if a user loads this page from `http://mail.mirapoint.com/wm/login.html`, he or she is redirected to `http://mail.mirapoint.com/extras/mail.mirapoint.com_std.html`

2. Create all your different **Login** pages. For each hostname, place a **Login** page in `/extras` with an appropriate name; for example the **Login** page you want people to see when they go to `ea.explore.mirapoint.com` should be `/extras/ea.explore.mirapoint.com_std.html`; use a standard edition WebMail **Login** page for a template. Or build your own custom **Login** page.
3. Associate the **Login** pages with delegated domains. In your new `ea.explore.mirapoint.com_std.html` **Login** page (for example) add a more JavaScript inside the existing `<script>` tag (if you used the Mirapoint **Login** page as a template, it will be there):

```
var myDomain = "@earlyaccess.mirapoint.com";
function appendDomain()
{
  document.loginform.user.value += myDomain; return true;
}
```

4. Continue customizing the **Login** page by finding the **Submit** button in your page; in the default `login.html` it looks like this:

```
<input type="submit" name="login" value="Login" class="genBtn">
```

Add an **onclick** attribute to that button that runs the **appendDomain** function when the user clicks **Submit**:

```
<input type="submit" name="login" value="Login" class="genBtn"
onclick="appendDomain()">
```

Short names will transform to long names when users log in. You can also rename the visible **User** option to something else, and make a hidden “user” field that gets populated by **appendDomain**, so that the user doesn’t see the username they typed in getting changed.

You can do exactly the same thing with Corporate Edition. Start with `/eml/login.html` and follow the same steps.



When you want one custom **Login** page per delegated domain, you can put a custom **Login** page in the delegated domain's `mail/login.html` brand and use the CLI command **Conf Enable Derivedomainurl** and it will achieve the same result (it will also transform short names to long names); however, the delegated domain has to match the end of the URL that was used to reach Webmail.

Branding the **Login** page using JavaScript is only necessary in the following cases:

- ◆ You are not using delegated domains, or
- ◆ You want to have multiple custom **Login** pages within a single delegated domain, or

- ◆ The hostname part of the URL used to reach the **Login** page does not end with the delegated domain's name (or using **Conf Enable Derivedomainurl** is impractical for some other reason).

Tips

This section offers some helpful brand development suggestions.

Create your brand primarily using the CSS files to simplify future brand upgrading.

A Custom Brand document is helpful when upgrading (see [Planning Your Brand/Creating a Brand Document](#) on page 12).

Do not do any of the following:

- ◆ Do not attempt to publish only a few files.
While entire directories can be deleted, the original file structure must be maintained in the zip file.
- ◆ Do not change the name of a file.
Your brand will break or the factory value will be used.
- ◆ Do not create any symbolic links.
Symbolic links are deleted during publishing.
- ◆ Do not attempt to publish zip files larger than 30 MB.
Files larger than 30 MB are rejected.
- ◆ Do not put custom JavaScript files in an application directory (for example, **mail**).
The system looks for custom JavaScript files in the **extras** directory and if they are in a different directory the custom JavaScript files will not be served.
- ◆ Do not mix components from different brands and/or different MOS releases.
Mixing causes failures. For example, using a **brandvars.var** or **css** file from a previous MOS fails when usage is attempted in the new MOS.

About Quotes

When defining variable keys (brandvars only) or values (most variables), all quotes inside of double quotes must either be single quotes, or they must be escaped with a backslash (\).

The following rules determine whether double quotes are required:

- ◆ If any character in the key (or value) is one or more of these:
 - ❖ a white space (space, tab, carriage return, line feed, etc.)
 - ❖ a reserved vartab (and vartab vectors) character: double quotes (“), equal sign (=), semicolon (;), parenthesis (), or comma (,)
- ◆ If either the key (or value) is of zero length.

A vartab vector describes a variable. For example:

```
msgfilterheaders =( "from","From",
                    ":tooc",  "To/CC",
                    ":subject", "Subject",
                    ":body",   "Body",
                    ":return-path", "Return-path",
                    ":x-junkmail", "X-Junkmail",
                    ":attachmenttype", "Attachment MIME Type",
                    ":attachmentfilename", "Attachment file name");
```

Locale display_charset Values

All textual content must be encoded in UTF-8. Don't change the "display_charset" value unless rendering in the browser as "ISO-8859-1" is preferred over the default "UTF-8." The encoding of the pages in a locale should match the cell.var variable "display_charset." These rules apply:

- ◆ Each application (since MOS 3.0+) has this line in it's cell.var:


```
display_charset = "<charset>";
```

 where "<charset>" is "Shift_JIS" for the ja_JP locale
and "UTF-8" for all other locales. Thus, each locale can have a different "display_charset."
- ◆ All "display_charset" values must be the same within a given locale as the data in the pages is transcoded accordingly and passed between applications with the expectation of consistent charset encoding.
- ◆ The encoding of the pages is expected to be in UTF-8 when the files are uploaded to the system (via Branding Publish). This is the baseline encoding.
- ◆ During run-time (or page-rendering time), the applications do the following with the <charset>:
 - ❖ Transcode the HTML pages to <charset>,
 - ❖ Set the HTML META charset to <charset>
 - ❖ Set the HTTP header to <charset>
 Browser responses to this have been observed to be:
 - ❖ MSIE browsers honor the HTML META charset exclusively
 - ❖ Netscape browsers honor the HTTP header exclusively



ASCII encoding is a proper subset of UTF-8. However, ISO-8859-1 encoding is not a proper subset of UTF-8.

In the event that there is an error during the page transcoding phase (for example, converting the UTF-8 page to ISO-8859-1), then the page is rendered as UTF-8; as a feature. Otherwise, truncated pages are displayed up to the point of the transcoding failure, which is not desirable.

Corporate Edition Branding

This chapter details the branding of the Corporate Edition version of the WebMail/WebCal graphical user interface (GUI); which can involve any of the following tasks:

- ◆ [Locating and Branding the Style Sheets](#)
- ◆ [Branding the Login Page](#) on page 33
- ◆ [Branding the Banner](#) on page 35
- ◆ [Branding Global Formats](#) on page 36
- ◆ [Branding the Tabs](#) on page 36
- ◆ [Branding the Links](#) on page 37
- ◆ [Branding the Options Links](#) on page 38
- ◆ [Branding the Calendar](#) on page 40
- ◆ [Locating and Branding the Help Files](#) on page 41

Branding details about the Standard Edition versions of the WebMail/WebCal GUI are in [Chapter 4, Standard Edition Branding](#) and details about Junk Mail Manager (JMM) are in [Chapter 5, Junk Mail Manager Branding](#).



Read [Chapter 1, About Branding](#) for important introductory information.

More information on style sheets is available in [Working with the Cascading Style Sheets](#) on page 20.

Locating and Branding the Style Sheets

Corporate Edition style sheets are located in `extras/css` and `extras/css/themes` directories of the system brand zip file (`sysbrand.zip`).

Corporate Edition cascading style sheets are used in the following order:

- ◆ `uui.css`: Styles defined the same across all themes.
- ◆ `themes/system-default.css`: Styles that may change.
- ◆ `themes/system-x.css`: Styles that do change for any of the several themes.

Located in `extras/css` directory, use the `ce_usage.html` example page to test your style sheet changes as you make them. Just make changes to the style sheets and open `ce_usage.html` in a browser to see those changes.

To see what the “theme” style sheets look like, log into Corporate Edition and go to **Options > Color Theme** (Figure 3).

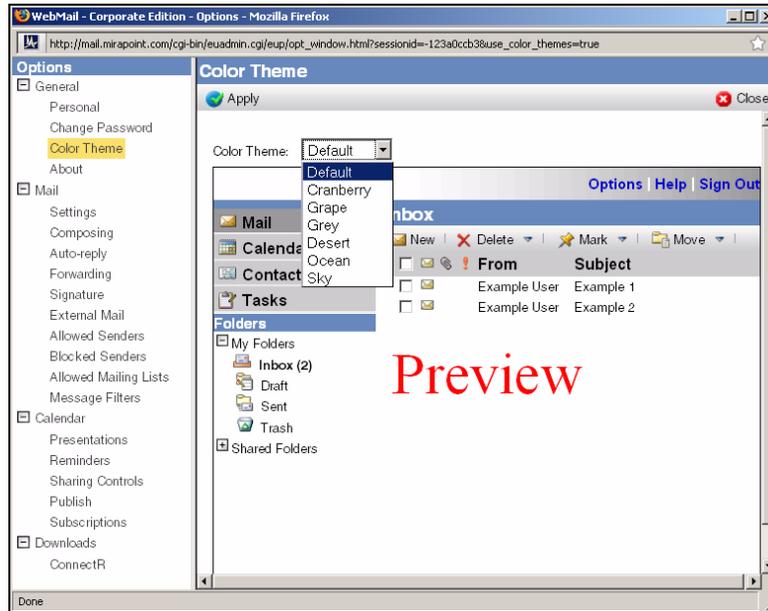


Figure 3 Options Pop-Up Window Example

For the Corporate Edition, the following themes and corresponding style sheet file names are available:

- ◆ **Default:** system-default.css
- ◆ **Cranberry:** system-5.css
- ◆ **Grape:** system-4.css
- ◆ **Grey:** system-2.css
- ◆ **Desert:** system-3.css
- ◆ **Ocean:** system-1.css
- ◆ **Sky:** system-6.css

The look of a theme can change by editing the style sheet file located in the `extras/css/themes` directory.

The display names for themes can also change, using the `mail/locale.var` file. For example: to change the “Cranberry” theme’s display name to “Summer”, in the `system-5.css` file located in the `mail/locale.var` directory, change the `Cranberry$` variable to `Summer$`.

```
"Cranberry"="Summer$(miravar_as_color_theme)";
```



Never change the style sheet file name.

To change the display name for a theme:

- ◆ Modify the value on the right-side of the equal sign in the following variables.

```
"Default"="Default$(miravar_as_color_theme)";
"Cranberry"="Cranberry$(miravar_as_color_theme)";
"Grape"="Grape$(miravar_as_color_theme)";
"Grey"="Grey$(miravar_as_color_theme)";
"Desert"="Desert$(miravar_as_color_theme)";
"Ocean"="Ocean$(miravar_as_color_theme)";
"Sky"="Sky$(miravar_as_color_theme)";
```

Branding the Login Page

This section describes how to change the images and text used on the Corporate Edition Login page (Figure 4).



Figure 4 Corporate Edition Login Page Example

Corporate Edition Login Images

To change the logo in the Login window:

- ◆ Replace the images found in the `images/eui` directory with your own image; but leave the image file names unchanged.
 - ❖ Ignore the “grey” versions of the GIFs, which set the grey fade on either side of the Login page.

Table 6 Image Files Used In the Login Window

File Name	Usage
gen_login_logo.gif	Logo graphic Example: Mirapoint®
gen_login_mailbox.jpg	Primary page graphic
gen_login_mailbox2.gif	Example: left hand side image of mailbox
gen_login_mailbox2.grey.gif	

Table 6 Image Files Used In the Login Window (Continued)

File Name	Usage
gen_login_grad.gif	Background black and white gradient
gen_login_grad.grey.gif	
gen_login_gradient.png	Upper gradient line (white to black)
gen_login_gradient2.png	Lower gradient line (black to white)

The gray bands at the top, the bottom and the background color of the **Login** window are defined in the `extras\css\uii.css` file. Edit the color or image source files for the following styles:

- ◆ `emlLogin`—for the page background color
- ◆ `emlLoginStrip1`— for the table background color
- ◆ `emlLoginStrip2`— for the black and white gradient bands above
- ◆ `emlLoginStrip3`— for the black and white gradient bands below
- ◆ Other images may be in the top-level `images/` directory.



The optimal image size for a replacement logo on the **Login** page is 113 x 30 pixels at 72 DPI. Other image dimensions may appear distorted.

Corporate Edition Login Text

To change the text on the **Login** page (mainly text appearance only), modify these variables in the `extras/brandvars.var` file.

- ◆ `"miravar_eui_title" = "$(eui_title)";` This variable defines the HTML “title” tag for the Login page.
- ◆ `"miravar_useStandardEdition" = "$(useStandardEdition)";` This \$variable provides the “Standard Edition” link, in a table. It uses the classes, `"emlLoginTitle"` (for “Corporate Edition”) and `"emlLoginEdition"` (for “Standard Edition”) for the text appearances; link styles use `"A.emlLoginEdition"` (for “Standard Edition”), defined in the `uii.css` style sheet.
- ◆ The **User** and **Password** option labels use the `"emlLoginLabel,"` `"emlLoginInput,"` `"emlLoginBtn,"` and `"emlLoginError"` classes defined in `uii.css`.
- ◆ `"miravar_eml_versionfooter_login" = "$(eml_versionfooter_login)";` This variable provides the following text, in a table: “Copyright © 1998-2008 [Mirapoint](#) Software, Inc. All rights reserved.” The word “Mirapoint” is a link. This variable uses the class `"emlLoginVersion"` and, for the link styles, `"A.emlLoginVersion"` defined in `uii.css`.
- ◆ `"miravar_useothermode" = "$(useothermode)";` This variable provides the **Cleartext | Secure** mode options and uses the `"emlLoginLink"` and `"A.emlLoginLink"` classes defined in `uii.css`.

- ◆ You cannot change the locale footer, except through its css classes, “emlOtherLocale” and “A.emlOtherLocal” defined in uui.css.
- ◆ "miravar_left_indent" and "miravar_right_indent" = "\$(left_indent)" and "\$(right_indent)" respectively. These variables provides border specifications.



You may also make changes directly to the login.html file located in **eml/locale/locale name/**.

Branding the Banner

This section describes how to change the images and text used on the Corporate Edition banner (shown in [Figure 5](#)).

- ◆ To change the logo in the banner, replace this image found in the **images/eui** directory: **gen_banner_logo.gif**.



Figure 5 Corporate Edition Banner Example



The optimal image size for a replacement banner logo is 113 x 30 pixels at 72 DPI. Other image dimensions may appear distorted.

- ◆ To change the background color, modify the genbanner class in the **extras/css/themes/system-*.css** files (by default this is set to use an image file). Link styles are also defined there.
- ◆ To add a link, other text, image, or insert custom content, edit the html file **/eml/locale/en_US.ISO_8859-1/banner.html**.
- ◆ To change the “Welcome” message colors, modify the genwelcome class in the **extras/css/themes/system-*.css** files.
- ◆ To change the banner height, use the **\$(eui_bannerheight)** variable in the **extras/brandvars.var** file. For example:

Original:

```
#default value
#eui_bannerheight = "100";
# Used by: Corporate Edition
"miravar_eui_bannerheight" = "$(eui_bannerheight)";
```

Modified:

```
#default value
#eui_bannerheight = "100";
# Used by: Corporate Edition
"miravar_eui_bannerheight" = "100";
```

Branding Global Formats

Date formatting, as it appears in the current WebMail and WebCal graphical interface, by default displays in the American style **month/day/year**.

To change the default American date format to the European date format, use the `miravar_date_entry_format` variable within the `extras/brandvars.var` file to change the variable value `%m/%d/%Y` (output sample: 02/15/2008), to the value `%d/%m/%Y` (output sample: 15/02/2008.) For example:

```
miravar_date_entry_format="%d/%m/%Y";
```

Branding the Tabs

The standard Mirapoint tabs identified within the `extras/brandvars.var` file are:

- ◆ `$(mail_tab)`
- ◆ `$(calendar_tab)`
- ◆ `$(contacts_tab)`
- ◆ `$(tasks_tab)`

Use the tabs variables as described in the `ce_usage.html` file, located in `extras/css` directory, to add your own custom tabs, as well as change the colors, change the spacing height and change the tab height of the tabs (Figure 6).

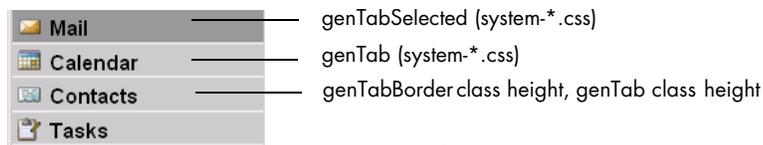


Figure 6 Corporate Edition Navigation Pane Example

- ◆ To adjust the spacing height and tab height, change the values in the `miravar_tab_info` variable. For example:

Original:

```
miravar_tab_info = "[8, 24]";
```

Modified:

```
miravar_tab_info = "[8, 12]";
```
- ◆ To add a custom tab, use the form `['sample', 'Sample Tab', '/images/image.gif', 'http://your_url.com', 'about:blank', false, false]`, whereby the fields are:
 - ❖ id prefix (must be unique)—For example: `'sample'`
 - ❖ label—for example: `'Sample Tab'`
 - ❖ icon—for example: `'/images/image.gif'`
 - ❖ rhs url—for example: `'http://your_url.com'`
 - ❖ links url—for example: `'about:blank'`
 - ❖ standard tab identification
 - “false”= not a standard Mirapoint tab
 - “true”= a standard Mirapoint tab

- ❖ calender dependency
 - “false”= no calendar needed
 - “true”= requires the calendar service to be started

Branding the Links

Use the “Navigation Pane Selectors” links variables as described in the `ce_usage.html` file, located in `extras/css` directory, to change the link styles, and selected vs. non-selected items (Figure 7).

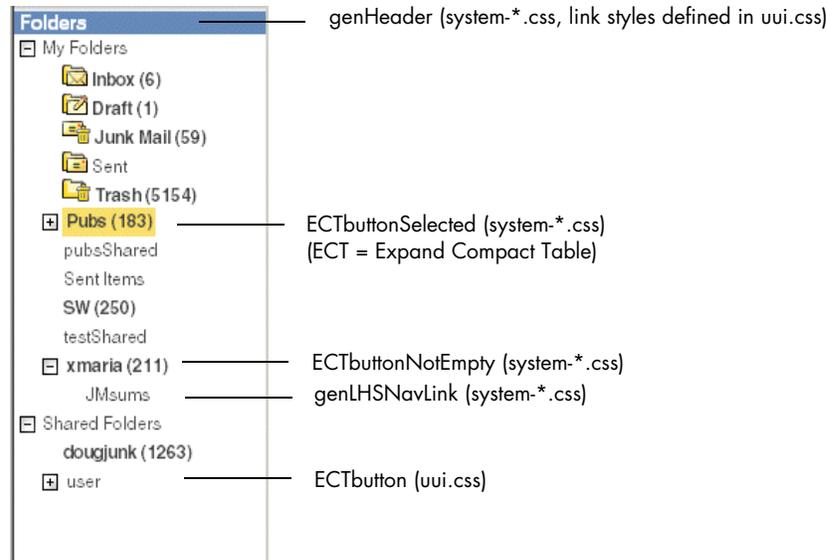


Figure 7 Corporate Edition Links Frame Example

- ◆ To hide the “Shared Folder” directory tree, remove the comment symbol in front of the `miravar_shared_folder_hide` variable in the `extras/brandvars.var` file. For example:

Original:

```
#key: miravar_shared_folder_hide
#desc: uncomment this variable in order to hide the entire "Shared Folder"
#tree view in CE.
# miravar_shared_folder_hide = "1";
```

Modified:

```
#key: miravar_shared_folder_hide
#desc: uncomment this variable in order to hide the entire "Shared Folder"
#tree view in CE.
miravar_shared_folder_hide = "1";
```

Branding the Options Links

All of the left hand side links in the **Options** pop-up window are modifiable within the `extras/brandvars.var` file (Figure 8). You can remove any of the links by deleting the corresponding link variable found in the `brandvars.var` file. For example: To prevent users from accessing the **Change Password** page, remove the `$(changepassword_link)` variable from the `miravar_options_general_preference_links` list.



Branches cannot be removed, nor can any of the branches or links be renamed.

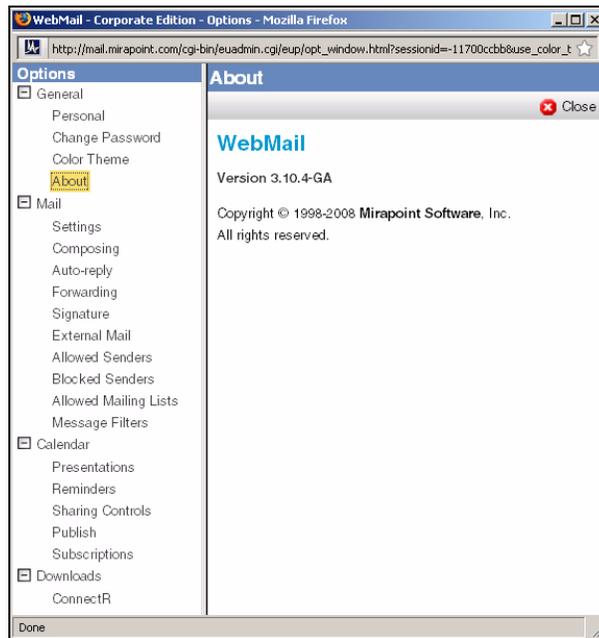


Figure 8 Corporate Edition Options Pop-Up Window

To remove an Options link, delete any of the following variables:

- ◆ **General** branch, `miravar_options_general_preference_links`:
 - ❖ `$(personal_link)`
 - ❖ `$(changepassword_link)`
 - ❖ `$(color_theme_link)`
 - ❖ `$(aboutpage_link)`
- ◆ **Mail** branch, `miravar_options_mail_links`:
 - ❖ `$(settings_link)`
 - ❖ `$(composing_link)`
 - ❖ `$(autoreply_link)`
 - ❖ `$(forward_link)`
 - ❖ `$(signature_link)`
 - ❖ `$(getmail_link)`
 - ❖ `$(whitelist_link)`
 - ❖ `$(blacklist_link)`

- ❖ `$(whitelistto_link)`
- ❖ `$(filter_link)`
- ◆ **Calendar** branch, *miravar_options_calendar_links*:
 - ❖ `$(presentation_link)`
 - ❖ `$(reminder_link)`
 - ❖ `$(sharing_link)`
 - ❖ `$(publish_link)`
 - ❖ `$(subscribe_link)`
- ◆ **Download** branch, *miravar_options_download_links*:
 - ❖ `$(downloads_MCMLink)`
 - ❖ `$(downloads_ConnectRlink)`
 - ❖ `$(downloads_link)`

Branding Individual Fields of an Option Page

In addition to the ability to hide whole **Option** page links, some fields on Option pages can be hidden by modifying the `extras/css/uuu.css` file. Hidden fields will not display on the right hand side of the Options pop-up window.

Hiding a field requires the uncommenting of the field name as listed in the `uuu.css` file. For example: to hide the *Email Address*: field on the *Mail > Composing* Options page, uncomment the `.options_compose_header_email_address` field name value in the `uuu.css` file.

Original:

```
/* .options_compose_header_email_address {display: none} */
```

Modified:

```
.options_compose_header_email_address {display: none}
```

Hiding the entire section on the Option page requires the uncommenting of the section header name as well as all of the sub-section fields, where applicable. For example: to hide the whole section called *Display* on the **Mail > Settings** Options page, uncomment everything for the name value `.options_settings_display_` in the `uuu.css` file.

Original:

```
/*.options_settings_display_header { display: none } */
/* .options_settings_display_perpage { display: none } */
/* .options_settings_display_recentlyreceived { display: none } */
/* .options_settings_display_messagepane_rows { display: none } */
/* .options_settings_display_showhtmlcontent { display: none } */
```

Modified:

```
.options_settings_display_header { display: none }
.options_settings_display_perpage { display: none }
.options_settings_display_recentlyreceived { display: none }
.options_settings_display_messagepane_rows { display: none }
.options_settings_display_showhtmlcontent { display: none }
```

The following page sections, fields and selectors can be hidden as needed:

- ◆ **General > Personal** page
 - ❖ Any of the fields or selectors.
- ◆ **Mail > Settings** page
 - ❖ Any or all of the **Display** section fields or selectors.
- ◆ **Mail > Composing** page
 - ❖ Any or all of the sections fields or selectors.
- ◆ **Calendar option > Presentations** page
 - ❖ Any of the selectors.

Branding the Calendar

This section describes how you can modify the New Event functionality used in the Corporate Edition of WebCalendar.

Choose a Resource

Resource searching is used for group calendar event scheduling to query available conference rooms and other defined resources. The resources are retrieved from the `groupcalendar:resourcelookup` URL, which is an LDAP query definition. The resources must be defined before the drop-down menu displays. The “Choose a resource” drop-down menu is automatically enabled and appears in the Corporate Edition **New > Event > Schedules** window using the `miravar_resource_search_enabled` variable.



Figure 9 Choose a Resource Drop-Down Menu Example

- ◆ To disable the drop-down menu, change the `miravar_resource_search_enabled` variable value in the `extras/brandvars.var` from “1” to “0”. For example:

Original:

```
miravar_resource_search_enabled="1";
```

Modified:

```
miravar_resource_search_enabled="0";
```

Large Group Event Scheduling Limit

(New feature in MOS 3.10.7/4.1.7) The Corporate Edition **New > Event > Schedules** window is used for group calendar event scheduling. The `miravar_large_group_threshold` variable determines how many users can be invited

to any one event scheduling. The default invitation limit is 100 users. The default limit can be changed in the `extras/brandvars.var` file.

- ◆ To change the limit in the `extras/brandvars.var`, change the `miravar_large_group_threshold` variable value. For example:

Original:

```
miravar_large_group_threshold="100";
```

Modified:

```
miravar_large_group_threshold="200";
```

Locating and Branding the Help Files

The Corporate Edition Help file pop-up window can also be branded (Figure 10).

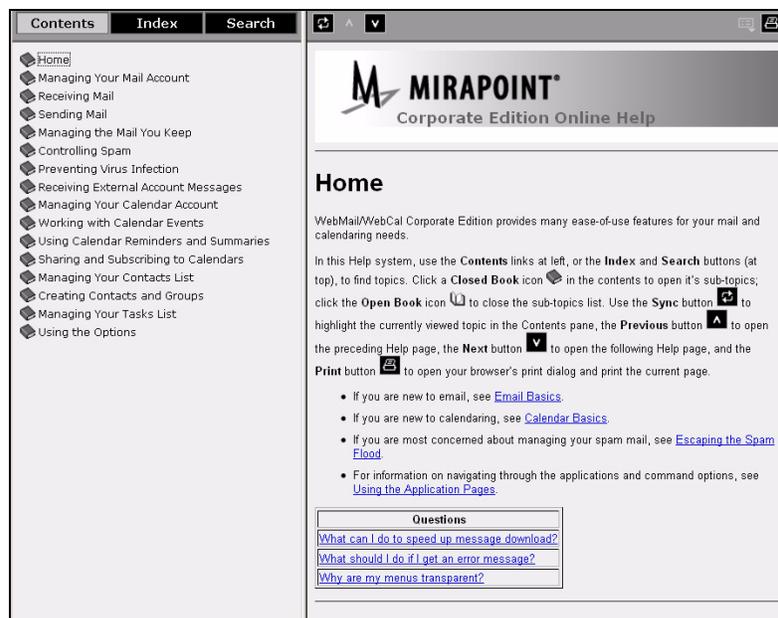


Figure 10 Corporate Edition Help File Sample

To brand CE Help files:

1. Locate the HTML files within the `sysbrand.zip` directory.
 WebMail and WebCal Help Files—`help/locale/locale/eml_I_Help/Output/`
 Only WebMail Help Files—`help/locale/locale/eml_W_Help/Output/`



Help file images are located in the `../Output/images` subdirectory.

2. Making brand style sheet edits using the `../Output/catalog.css` and `../Output/document.css` files.



Avoid deleting the paragraph tagging, which affect cross-reference links. For example: “`a name=wp1076530`”

Next Steps

Once you have branded the Corporate Edition files, you must zip the files and publish them. See [Publishing a Brand](#) on page 103 for details. Additionally, you can access the Administration Suite online help for more information. Go to **System > Branding > Help**.

Standard Edition Branding

Standard Edition branding consists of changing any of the files that make up the available “end-user applications,” which currently consist of: WebMail, WebCal, Options, Spelling, and Address Book. Much of the Standard Edition branding and Junk Mail Manager branding is done similarly, but separately (see [Chapter 5, Junk Mail Manager Branding](#)).

Branding details about the Corporate Edition version of the WebMail/WebCal GUI are in [Chapter 3, Corporate Edition Branding](#).



Read [Chapter 1, About Branding](#) for important introductory information.

This chapter describes the files available for Standard Edition branding and some application-specific branding tasks, including the following tasks:

- ◆ [Common Standard Edition Branding](#)
- ◆ [Branding the WebMail Direct Pages](#)
- ◆ [More WebMail Branding](#), includes these tasks:
 - ❖ [Setting a Default Charset for WebMail](#)
 - ❖ [Branding Default Mode Preferences for WebMail](#)
 - ❖ [Changing the Date Format in WebMail](#)
 - ❖ [Changing How WebMail Handles Attachments](#)
- ◆ [Branding the WebCal Direct Pages](#)
- ◆ [More WebCal Branding](#), includes these tasks:
 - ❖ [Supporting Group Calendar](#)
 - ❖ [Branding Default Mode Preferences for WebCal](#)
 - ❖ [Branding Calendar Email Messages](#)
- ◆ [Branding the Options/Account Pages](#)
- ◆ [Branding the Spelling Pages](#)
- ◆ [Branding the Address Book Pages](#)
- ◆ [More Address Book Branding](#), includes these tasks:
 - ❖ [Localized Import/Export in Address Book](#)
 - ❖ [Branding Contact Data](#)
 - ❖ [Adding LDAP Search Fields](#)

◆ Next Steps



There are many variables, hidden input types, and code that you should not change. Doing so may break your application. This chapter describes the elements, and variable values that you can change and offers guidelines for HTML changes that you can make.

Common Standard Edition Branding

This section describes the most common branding tasks available for Standard Edition.

Detailed information on all the files that make up WebMail/WebCal Standard Edition and additional available branding tasks, are provided in the following sections, beginning with [Branding the WebMail Direct Pages](#) on page 48.

Branding the Style Sheets

The Standard Edition style sheet, `style.css`, is located in the `extras/css` directory of the system brand zip.

Located in `extras/css` directory, use the `usage.html` example page to test your style sheet changes as you make them. Just make changes to the style sheets and open `usage.html` in a browser to see those changes.

Find more information about style sheets in [Working with the Cascading Style Sheets](#) on page 20.

Branding the Login Page

You can customize the Standard Edition WebMail, WebCal `login.html` file with specified colors, a different logo, and links ([Figure 11](#) on page 44). This section describes changing the WebMail Login page defined in the `mail/login.html` file. You can follow these same procedures to change the WebCal Login page defined in `mcal/login.html` file.

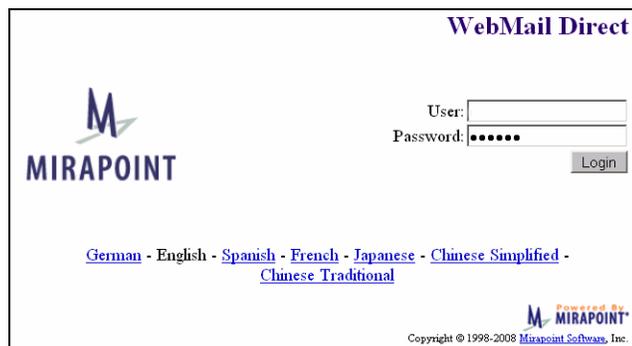


Figure 11 Standard Edition WebMail Direct Login Page Example

Standard Edition Login Images

To change the logo, located in **mail/images**, and **mcal/images** directories, respectively, replace this image: **logosmall.gif**. This logo has width=142 height=118.



Do not remove or replace the **poweredby.gif**. The **poweredby.gif** logo should be left on the **Login** page. There are some options as to which **poweredby.gif** you use, see [Powered by Mirapoint](#) on page 26 for details.

To change the background colors, change the **style.css** selectors “**genLogin**” (text and background color) and “**genLoginTitle**” (text only) to the colors you want.

You can add a background image using the HTML **background** tag to the **body** class rule; for example:

```
<body class="genLogin"
onload="document.forms[0].elements[0].focus()"$(miravar_genLogin)
background="/images/login_background.gif">
```

Maintain the fallback so that your specified colors display even in a non-CSS enabled browser, open the **brandvars.var** file using a text editor and replicate your declarations as the appropriate values. For example, if you change the “**genLogin**” background in to light blue (**#ccffff**) also change the **\$(miravar_genLogin)** **bgcolor** value in **brandvars.var** to **#ccffff**. The **brandvars.var** variables **\$(miravar_genLogin)** and **\$(miravar_genLoginTitle)** are fallbacks for “**genLogin**” and “**genLoginTitle**”.

Standard Edition Login Text

The “**User**” and “**Password**” text are defined in **mail/login.html**, and **mcal/login.html** respectively. You can change those words, but do not change the login form (also defined there).

The locale version footer is defined based on the locales that you have added to your system (through the CLI). You cannot change the names of the locales, but you can use “**genLogin**” in the **extras/css/style.css** to change the colors, font, and background colors.

To add a link:

1. Create the introduction/information web page that you want to be available to people from the **Login** page and save it to the top-level **extras** directory.
2. Open a text editor such as MS Notepad and open the desired **login.html** or **rg_login.html** file.
3. Create a new link by adding an **href** tag referencing your non-system web page.

Branding the Banner

The Images and text used in the Standard Edition banner ([Figure 12](#)) can change using the **mail/banner.html** file.



Figure 12 Standard Edition Banner Example

The following branding variables (Table 7) and procedures apply to the WebCal banner, using the `mcal/banner.html` file.

Table 7 banner.html Variables

	Variable	Defined in...	Comments
BODY class	genNav	CSS and brandvars.var	Text and background colors
Apps Menu	\$(appsmenu)	cell.var (key: appsmenu_tb)	<ul style="list-style-type: none"> ❖ WebMail and WebCal both licensed ❖ There is a placeholder for a logo
	genDataOff	CSS and brandvars.var	Disabled links text color
	genNavLink		Colors for regular, active, visited, and hover. Text-decoration : none.

- ◆ To change the background color, modify the “genNav” class in the `extras/css/style.css` file. Link styles, “genDataOff” and “genNavLink”, are also defined in the `extras/css/style.css` file.
- ◆ To add an image, place the image file you want to add in the `mail/images`, `mcal/images`, or `spam/images` directory, respectively. At the top of the `banner.html` file, beneath the HTML `body` class rule, add the `img src` tag as follows:

```

```



Use `/wmi` as an alias to `/mail/images`.

- ◆ To scale the image, specify `width` and `height` variables within the `img src` tag. For example:

```

```

Branding the Links Frame

The Links frame (Figure 13) uses the `mail/links.html`, `mcal/links.html`, and `spam/links.html` files; and, displays on every WebMail page when using **Frames** or **Frames with JavaScript** modes. The **No Frames** mode defines the link elements in the template files using a `$(simplelinks)` variable.



Figure 13 WebMail Standard Edition Links Frame Example

Links frame branding involves the use of the following variables (Table 8):

Table 8 links.html Variables

	Variable	Defined in...	Comments
BODY class	genNav	CSS and brandvars.var	Text and background colors
Primary Links	N/A	links.html	HREF links (except as noted below); uses <small> tag
	\$(compact_link) \$(emptytrash_link) \$(getmaillink)	cell.var (system key)	The system turns on or off these links depending on preference settings
	\$(helppath)	cell.var (system key)	Path to the online Help files
	genNavLink	CSS and brandvars.var	Colors for regular, active, visited, and hover. Text-decoration : none
Legend	N/A	links.html	“Legend” text font size=-1 Icon label text font size=-2
	\$(legend_priority)	cell.var (key: prioritycount_3or5)	How many Priority symbols display
	\$(deletelegend)	cell.var (key: deletelegend)	If Trash is turned off, this Legend displays with the Delete icon
	genHeader	CSS and brandvars.var	Header row color for table
	genLegend		Background color for table

Adding a Linked Advertisement

You can add a graphic image to your pages to advertise another Web site, providing direct link access. This procedure describes how to add a linked advertisement to the list of primary links defined in the links.html file in the **mail** directory for WebMail. You can follow this same procedure for the links.html file in the **mcal** directory for Calendar. You can also apply this procedure to the banner.html file

To add an advertisement to that file instead of the links.html file:

1. Place the advertisement image file in the appropriate **images** directory.



Use **/wmi** as an alias to **/mail/images**.

2. At the top of the links.html file, beneath the HTML **body class** rule, use an **img src=** tag in conjunction with an **href=** tag, as shown in the following example. Specify a URL that points to the web server where the rotating images reside:

```
<a href="http://www.webserver/image_name.html">
</a>
```

Substituting Button Images for Text Links

This procedure describes how to substitute button .gif images for the linked text defined in the `links.html` file in the `mail` directory for WebMail. You can follow this same procedure for the `links.html` file in the `mcal` directory for Calendar.

1. Place the .gif files for the buttons into the appropriate `images` directory.
2. In the `links.html` file, locate the lines of text between the `<small>` and `</small>` tags. These lines provide links to their respective WebMail windows and panes. The first line in this series specifies checking for new mail:

```
<small>
  <a class="genNavLink" href="/wm/mail/
  checkmail.html?sessionId=$(sessionId)&op=checkmail&mbox="
  target="rhs">Check Mail</a>
</small>
```

3. To add a button image for checking new mail, substitute the following for `Check Mail` in the line.

```

```

4. Repeat for each button image you want to use.

Branding the Help Files

Locate the HTML files and make whatever changes you want.

The Standard Edition files are here:

`help/locale/locale name/addrBookCom/`: Address Book Help files used by both WebMail and WebCal.

`help/locale/locale name/images/`: WebMail/WebCal Help image files.

`help/locale/locale name/user/`: WebMail Options and Account Administration Help files.

`help/locale/locale name/WebMail/`: WebMail Help files.

Standard Edition Help files use the `style.css` in the `extras/css` directory.



The URL that is shown at the bottom of the browser window when an online help page is displayed is: `/help/apps/locale`. Do not create a directory to mimic the hierarchy shown in the browser window. Keep the structure of the original zip file: `help/locale/locale name/`.

Branding the WebMail Direct Pages

The files that create the WebMail SE pages are located in the `mail` directory. The pages are created by a combination of process files that manage requests, and HTML files that provide templates for the display. The HTML template files reference a cascading style sheet (CSS). The style elements defined in the CSS are also defined in a special process file, `brandvars.var`, so browsers that do not support CSS still display specified style formatting.

WebMail Addressbook

WebMail Address Book uses the same files WebCal Address Book uses. Dollar sign (\$) variables are used to call out the differences between the two applications. Addressbook branding is discussed in [Branding the Address Book Pages](#) on page 84.

Planning Your Custom Brand

The following interface elements can be branded:

- ◆ **Graphics:** You can add graphics to the **Login** page, the Banner and Links frames, and change the graphics used for icons. New graphics can be added to the top level **images** directory or to the individual application **images** directories. **Note:** You may need to change frame sizes if you change or add graphics; frame sizes can be specified as percentages or fixed widths. WebMail contains the following graphics located in the **mail/images** folder (“wmi” in the templates is an alias for **mail/images**):
 - ❖ **Login graphic:** This Mirapoint logo is referenced in the login.html file.
 - ❖ **Legend graphics:** The Legend is a table defined in links.html and references graphics in the **mail/images** directory.
 - ❖ **Inbox message flag graphics:** These graphics are referenced in the toc.html file.
 - ❖ **Preferences Mode graphics:** These graphics are referenced in the prefs.html file.
- ◆ **Colors:** You can change the background, contrast, and text colors throughout the web pages. You can change the background and header colors for most of the files in the CSS.
- ◆ **Terminology:** You can change the names of the links, page titles, and commands. For example, you can change **Preferences** to **Settings** in WebMail and the corresponding online Help pages and error messages. For example; if you change **Preferences** to **Settings**, all online Help files describing the **Preferences** page should also be changed to use **Settings**.
- ◆ **Arrangement of information:** You can change the order of the columns in the **Inbox** page. You can also change the order of the links and commands, the position of links, buttons, fields, text, and so forth. **Note:** If you re-arrange data columns, you may need to adjust the colspan definitions (the number of columns spanned by a cell) for the table, usually part of a variable defined in cell.var. This is because some data columns may require a colspan greater than others. You will also need to adjust the positions of the column headers in the template file.
- ◆ **Functionality:** You can add or remove functionality. You can comment out a feature that you do not want to support, or you can add off-box functionality of your own. For example, you can add a link to a page that describes your newly branded WebMail.

WebMail Viewing Modes

WebMail provides users with three different viewing mode options:

- ◆ **Frames**—pages display in three frames; “banner,” left-side “links,” and “template.”
- ◆ **Frames with JavaScript**—Similar to the **Frames** mode with the addition of a “TOC” frame and JavaScript that allows you to view the message list and a message on the same page.
- ◆ **No Frames**—pages display without the use of frames.

Users select a mode in the **Preferences** page of WebMail or WebCal. **Frames** is the default mode, and can change with branding.

To hide modes from users:

- ◆ Comment out the other modes on the **Preferences** page (prefs.html).

To show users branding on all three modes:

- ◆ Brand all three modes, which involves branding additional files.

Frames Mode

The **Frames** mode has no bottom frame to display messages, which results in your message (read_s.html) displaying in the frame that previously displayed the message list (toc.html)([Figure 14](#) on page 51). A frame definition file, window.html, defines a frameset that references the **banner** (banner.html), **links** (links.html and blank.html) and **checkmail** frames (checkmail_s.html) ([Table 9](#)).

Table 9 Frames Mode Example

Description	Example
<p>The frameset is defined in windows.html. The top frame is created by banner.html; links.html and blank.html create the left frame.</p> <p>This mode often uses the same files as the No Frames mode.</p>	

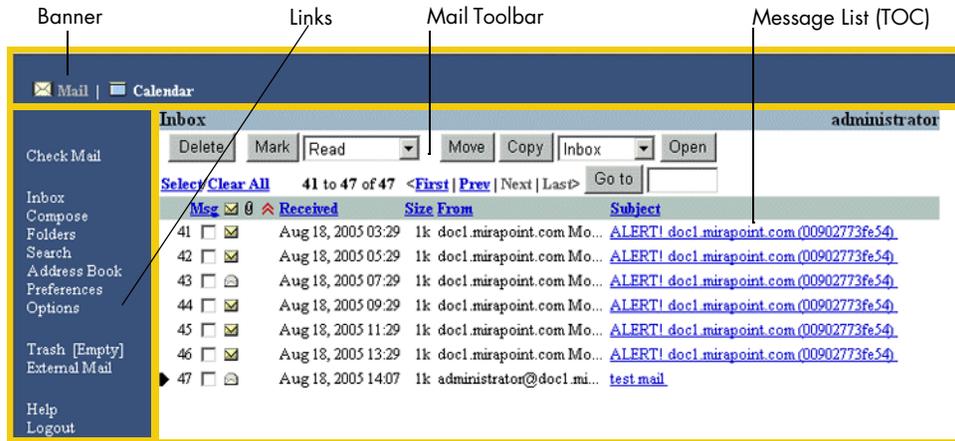


Figure 14 WebMail in Frames Mode Example

The TOC elements are defined in the `checkmail_s.html`. When you click on a message link, the message (`read.html`) replaces the **checkmail** frame (`checkmail_s.html`).



For the **Frames** mode, the **checkmail** frame references the `checkmail_s.html` file, not the `checkmail.html` file, which is the frameset definition file used by the **Frames with JavaScript** mode.

Frames with JavaScript Mode

In the **Frames with JavaScript** mode the TOC frame expands vertically to display your list of messages, and the bottom frame displays a message (`read.html`) when you click on a message link (Table 10).

Table 10 Frames with JavaScript Mode Example

Description	Example
<p>The frameset is defined in <code>windows.html</code>. The top frame is created by <code>banner.html</code>; <code>links.html</code> and <code>blank.html</code> create the left frame.</p> <p>This mode sometimes uses the same files as the No Frames mode.</p> <p>You can add JavaScript to all three modes if you want, but the Frames with JavaScript mode requires it for certain functions.</p>	<p style="text-align: right;">Frameset = windows.html</p>

A frame definition file (`window.html`) defines a frameset that references the **banner** (`banner.html`), **links** (`links.html` and `blank.html`) and **checkmail** frames (`checkmail.html`). The **checkmail** frame (`checkmail.html`) defines another frameset that references the **toolbar** frame (`toolbar.html`) and `inner.html`, another frame definition file. The `inner.html` frame definition file defines a frameset that references

the TOC frame (toc.html), and the **template** frame, displays the message contents and commands; read.html when a message is selected and read_blank.html when no message is selected (Figure 15 on page 52).

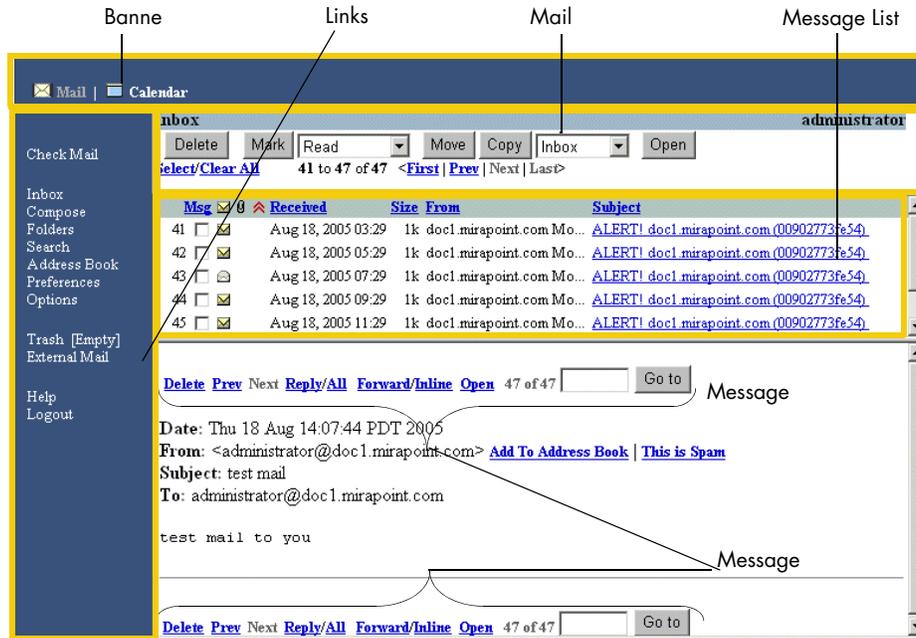


Figure 15 Frames With JavaScript Mode, WebMail Standard Edition

No Frames Mode

In WebMail, the **No Frames** mode (Table 11) has several files in common with the **Frames** mode, including:

- ◆ *checkmail_s.html*: Displays the Inbox or selected folder and performs a mail check
- ◆ *search_s.html*: Displays the Search page
- ◆ *read_s.html*: Displays your message

Table 11 No Frames Mode Example

Description	Example
No frameset is used. Sometimes called “simple”; files with an underscore+s (_s) are used by this mode and often by the Frames mode as well. An example is checkmail_s.html.	<div style="border: 1px solid black; width: 150px; height: 150px; margin: 0 auto; display: flex; align-items: center; justify-content: center;"> checkmail_s.html </div>

The **Frames** mode and **No Frames** mode can share these files because variables at the top of the files direct the system at run-time as to whether the file should display

in a frame or by itself with the simplelinks menu. Because the **No Frames** mode is designed to be very simple, it is often not branded (Figure 16). But you must brand those common files in order for the **Frames** mode, typically the default mode, to be branded. When you click on a message link the message (read.html) replaces *checkmail_s.html*.

For No Frames mode, a different Apps Menu and “simplelinks” menu display; shown below

For Frames mode, banner.html and links.html are used instead of the shown Apps Menu and

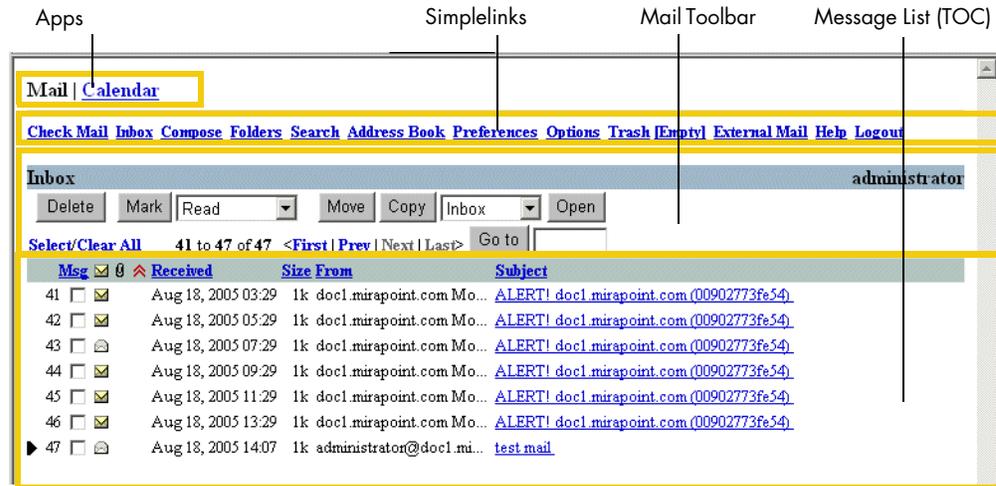


Figure 16 No Frames Mode, WebMail Standard Edition

About the WebMail Screen Elements

Throughout WebMail these elements either display in a frame file such as banner.html or links.html for the frames modes (**Frames** and **Frames with JavaScript**); or, for the **No Frames** mode, HTML in the files (checkmail_s.html, search_s.html, and read_s.html) creates them. Variables tell the system how to display the:

- ◆ **Apps Menu:** Only displays if WebMail Direct and WebCal Direct are both licensed. Active link changes depending on the application.
- ◆ **Primary Links Menu or Simplelinks:** Defined in links.html in a frame mode. Or as a variable (\$simplelinks) in the checkmail_s.html, search_s.html, and read_s.html files **No Frames**.
- ◆ **Toolbar and Status:** Defined in toolbar.html for **Frames with JavaScript** mode. Defined in checkmail_s.html, search_s.html, or read_s.html for the **Frames** and **No Frames** modes. The **Prev** and **Next** page links are part of the **toolbar** element.
- ◆ **TOC and Message List:** Defined for the **Frames with JavaScript** mode in toc.html; references the toc.js file that contains the JavaScript. In the **No Frames** and **Frames** modes, the TOC is defined in checkmail_s.html and search_s.html; the message list is defined in cell.var.
- ◆ **Message Contents and Commands:** Defined for the **Frames with JavaScript** mode in read.html or read_blank.html. In the **No Frames** and **Frames** modes, defined in read_s.html.

Frame Files Reference

This section provides a reference for each of the WebMail HTML template files used by the **Frames** and **Frames with JavaScript** modes to create the frames. Following this section is a section on the remaining page files used by the **No Frames**, and sometimes, the **Frames** mode.

banner.html

The Banner frame, banner.html (Figure 17), displays on every WebCal page when in **Frames** or **Frames with JavaScript** modes. **No Frames** mode has these elements defined in the template files with a `$(appsmenu)` variable. Table 12 describes elements and variables.



Figure 17 Banner, Calendar highlighted

Table 12 banner.html Variables

	Variable	Defined in...	Comments
BODY style	genNav	CSS and brandvars.var	Text and background colors
Apps Menu	\$(appsmenu)	cell.var (key: appsmenu_tb)	<ul style="list-style-type: none"> ❖ WebCal and WebMail both licensed ❖ There is a placeholder for a logo
	genDataOff	CSS and brandvars.var	Disabled links text color
	genNavLink		Colors for regular, active, visited, and hover. Text-decoration : none.
Banner height definitions	none	cell.var (key: bannerheight_noapps) (key: bannerheight_withapps)	These have to be modified for proper display of any image placed in the banner.

links.html

The Links frame, links.html, shown in [Figure 18](#), displays on every WebMail/ WebCal page when in **Frames** or **Frames with JavaScript** modes. **No Frames** mode has these elements defined in the template files with a \$(simplelinks) variable. [Table 13](#) describes elements and variables.



Figure 18 Links Frame, WebMail on the left, WebCal on the right

Table 13 links.html Variables

Element	Variable	Defined in...	Comments
BODY style	genNav	CSS and brandvars.var	Text and background colors
Primary Links — WebMail	N/A	template	HREF links (except as noted below); uses <small> tag
	\$(compact_link) \$(emptytrash_link) \$(getmaillink)	cell.var (system key)	The system turns on or off these links depending on preference settings
	\$(helppath)	cell.var (system key)	Path to the online Help files
	genNavLink	CSS and brandvars.var	Colors for regular, active, visited, and hover. Text-decoration : none
Primary Links — WebCal	N/A	template	<ul style="list-style-type: none"> ❖ <table cellpadding=0 cellspacing=0 border=0> ❖ HREF links (except as noted below) ❖ uses <small> tag
	\$(newevent_link) \$(todoitem_link) \$(schedmtg) \$(deleteevents_link) \$(importexport_link) \$(viewother_link) \$(addrbook_link) \$(prefs_link)	cell.var	The system turns on or off these links depending on licensing
	\$(helppath)		Path to the online Help files
	genNavLink	CSS and brandvars.var	Colors for regular, active, visited, and hover. Text-decoration : none.

Table 13 links.html Variables (Continued)

Element	Variable	Defined in...	Comments
Legend	N/A	template	“Legend” text font size=-1 Icon label text font size=-2
	\$(legend_priority)	cell.var (key: prioritycount_3or5)	How many Priority symbols display
	\$(deletelegend)	cell.var (key: deletelegend)	If Trash is turned off, this Legend displays with the Delete icon
	genHeader	CSS and brandvars.var	Header row color for table
	genLegend		Background color for table

toolbar.html (WebMail only)

The Toolbar frame, toolbar.html, shown in [Figure 19](#) with the Delete to Trash feature disabled (default) and in [Figure 20](#) with the Delete to Trash feature enabled, displays on the **Inbox** and **Checkmail** page for any selected folder, and the **Search** page, when in **Frames with JavaScript** mode. **No Frames** and **Frames** modes have these elements defined in the template files. [Table 14](#) describes elements and variables.

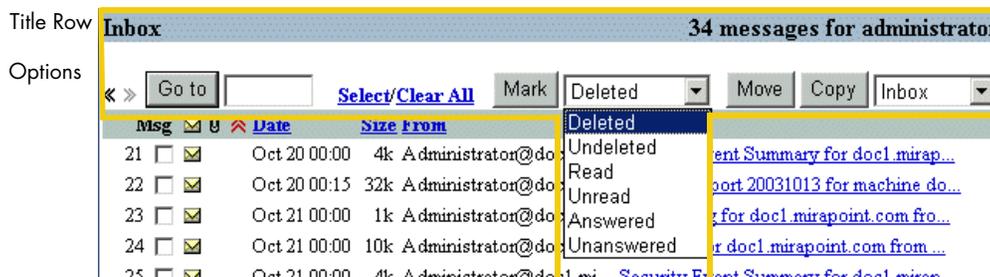


Figure 19 toolbar.html Frames with JavaScript Mode, Delete to Trash disabled

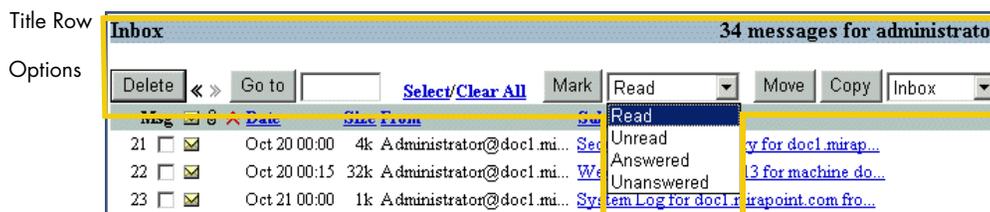


Figure 20 toolbar.html Frames with JavaScript Mode, Delete to Trash enabled

Table 14 toolbar.html Variables

Element	Variable	Defined in...	Comments
BODY style	\$(miravar_body)	CSS and brandvars.var	<ul style="list-style-type: none"> ❖ Text and background colors ❖ Active, Unvisited, Visited, and Mouse-over link colors
Frame margin	\$(framemargin)	cell.var (system key)	Margin space for the frame
Title Row	\$(foldername)	cell.var (system keys)	<ul style="list-style-type: none"> ❖ Name of selected folder ❖ Quota graph ❖ Number of messages ❖ Name of user ❖ Change order or text in template
	\$(quota_graph)		
	\$(msgcnt)		
	\$(user)		
	genTitle	CSS and brandvars.var	Text and background colors
Options			
All buttons	N/A	template	Change position or text in template
	genBtn	CSS and brandvars.var	Text and background colors
Approve button	\$(qtnapprovebuttonjs)	cell.var (key: qtnapprovebuttonjs)	<ul style="list-style-type: none"> ❖ Adds/removes Approve button ❖ Change position in template; change text in cell.var
Rescan button	\$(virusscanbuttonjs)	cell.var (key: virusscanbuttonjs)	<ul style="list-style-type: none"> ❖ Adds/removes Rescan button ❖ Change position in template; change text in cell.var
Delete button	\$(deltrashbuttonjs)	cell.var (key: deltrashbuttonjs)	<ul style="list-style-type: none"> ❖ Adds/removes Delete button ❖ Change position in template; change text in cell.var
Page links	\$(prevlink)	cell.var (keys: prevpjs_on, prevp_off)	Change position in template; change text in cell.var
	\$(nextlink)	cell.var (keys: nextpjs_on, nextp_off)	
Select/ Clear All links	N/A	template	<ul style="list-style-type: none"> ❖ Uses JavaScript ❖ Change position or text in template

Table 14 toolbar.html Variables (Continued)

Element	Variable	Defined in...	Comments
Mark options	\$(markdeletedjs)	cell.var (system keys)	<ul style="list-style-type: none"> ❖ These options change depending on preferences Trash setting ❖ Change position or text in template ❖ System database supplies this user input data on demand
	\$(delete_sel)		
	\$(undelete_sel)		
	\$(read_sel) \$(unread_sel)		
	\$(reply_sel) \$(unreply_sel)		Defined by the user Delete to Trash preferences setting; adds/removes Deleted and Undeleted options
\$(usetrash)			
Move and Copy options	\$(mboxlistopt)	cell.var (system key)	<ul style="list-style-type: none"> ❖ Folder drop down list ❖ Change position or text in template

toc.html (WebMail only)

The Table of Contents frame, toc.html, shown in [Figure 21](#), displays on the **Inbox** and **Checkmail** page for any selected folder, and the **Search** page in **Frames with JavaScript** mode. **No Frames** and **Frames** modes have these elements defined in the template files. [Table 15](#) describes elements and variables.

Icon Graphics

Header Row	Msg	Date	Size	From	Subject
Message List	1 <input type="checkbox"/> <input checked="" type="checkbox"/>	Jul 16 17:04	1k	doc1.mirapoint.com Mo...	ALERT! doc1.mirapoint.com (00902773fe54)
	2 <input type="checkbox"/> <input checked="" type="checkbox"/>	Jul 16 17:05	1k	doc1.mirapoint.com Mo...	ALERT! doc1.mirapoint.com (00902773fe54)
	3 <input type="checkbox"/> <input type="checkbox"/>	Jul 16 17:06	1k	doc1.mirapoint.com Mo...	ALERT! doc1.mirapoint.com (00902773fe54)
	4 <input type="checkbox"/> <input checked="" type="checkbox"/>	Jul 16 18:10	1k	nobodv@doc1.mirapoint	Scheduled pattern file update

Figure 21 toc.html Frames with JavaScript Mode

Table 15 toc.html Variables

Element	Variable	Defined in...	Comments
BODY style	\$(miravar_body)	CSS and brandvars.var	<ul style="list-style-type: none"> ❖ Text and background colors ❖ Active, Unvisited, Visited, and Mouse-over link colors
Header Row	genHeader	CSS and brandvars.var	Text and background color
	genHeaderLink		Link style and color
Text	N/A	template	<ul style="list-style-type: none"> ❖ To re-position columns, see also Message List element, below. ❖ Font size=-1
Graphics	N/A	/images directory	unread.gif, attachment.gif, and priority.gif

Table 15 toc.html Variables (Continued)

Element	Variable	Defined in...	Comments
Sorted on icon (^ and v)	\$(msgorderimage) \$(seenorderimage) \$(attachorderimage) \$(priorityorderimage) \$(dateorderimage) \$(sizeorderimage)	system defined	Display the Ascending/Descending Sorted on icons (^/v)
Graphics' ALT text	N/A	template	Change order or text in template
Message List	\$(toc)	cell.var (key: toc_)	Creates the message list table rows of data. To re-position the columns, you must change the text in the template (Header Row Text element above) and change the order of the data in this variable.
		cell.var (key: toc_empty_subject) (key: toc_empty_sender)	<ul style="list-style-type: none"> ❖ Text to display in the TOC if the message Subject: header is empty ❖ Text to display in the TOC if the message From: header is empty



Do not change the JavaScript at the bottom of the page or the page will break.

How the WebMail Pages are Defined

This section describes the frame, frameset, process (vartab and so forth), and HTML template files that define the pages. Some CSS selectors defined in style.css are used on every page. Refer to [Working with the Cascading Style Sheets](#) on page 20 for details on the CSS.

WebMail Frameset Definition Files

[Table 16](#) describes the WebMail frame definition files. These files are only used in the **Frames** and **Frames with JavaScript** modes. You may need to adjust the frame set definitions in these files if you reorder elements or add graphics.

Table 16 WebMail (mail/) Frameset Definition Files

Template File	Description & Dependencies
blank.html	A frame definition file that ensures that the correct files display in the correct order.
checkmail.html	References toolbar.html and inner.html for the Inbox (or any selected folder).
inner.html	References toc.html and read.html, or read_blank.html when no message is selected, for the Inbox (or any selected folder).

Table 16 WebMail (mail/) Frameset Definition Files (Continued)

Template File	Description & Dependencies
search.html	References searchform.html, toolbar.html, and inner.html for the Search page.
window.html	Controls which files are displayed and how much room they take up. References banner.html, links.html and blank.html

WebMail Process Files

Table 17 describes the WebMail process files that are used to process JavaScript and system requests. You can make global changes by changing variable definitions in these files.

Table 17 WebMail (mail/) Process Files

	Description & Dependencies
blank_page.html	Used by the system. Do not change.
cell.var	Contains variable definitions.
charsets.var	Contains character set name definitions.
html_compose.js	Contains the JavaScript that the compose.html uses in the HTML Text mode.
locale.var	Contains customizable system and error messages.
rfc822.txt	Contains a dollar sign (\$) value.
timezones.var	Contains timezone name definitions.
toc.js	Contains the JavaScript that the checkmail.html and search.html files use in the Frames with JavaScript mode.



To add your own JavaScript files to an application, they must be placed in the **extras** directory.

About cell.var

The cell.var file contains definitions for many of the screen elements including the rows of data in the tables. The HTML template files contain variables that are replaced by dynamically generated table rows; the formatting, positioning, data, and so forth. A vartab file generates the contents that replace the table variables using key:value pairs; for example:

```
mbox_row = "<tr> <td> $(username) </td> </tr>";
```

In the example above, the key name is **mbox_row**, and the value is everything within the double quotes on the right of the equal sign (=), which defines a table row of data containing the variable \$(username). The variable used in the example,

\$(username), is a system variable that only the system can define; it is not defined in any vartab file.



These rules apply to all of the vartab files.

Reading the cell.var File Comments

In the cell.var file of each application, there are many variable key:value pairs that you can modify. There are also system variables and JavaScript that should be left alone.

Each of the cell.var files have been commented to make it easier for you to understand what you can modify and how. The comments use a simple pound sign (#) along the left margin to distinguish them and are usually placed right above the actual corresponding code. **Note:** The comment character must be in column 1 outside of the key:value pair. The following is an example:

```
#key: prioritycount_3or5
#desc: Number of priority options.
#3 - 'High', 'Normal' and 'Low'
#5 - 'Highest', 'High', 'Normal', 'Low' and 'Lowest'
prioritycount_3or5 = "5";
```

The example translates as follows:

“key:” is the key name, what you see on the left of the equal sign (=) of a key:value pair.

“desc:” is a description of the variable.

Possible values for the variable follow.

The actual code used comes last; in the given example...

```
prioritycount_3or5 = "5";
```

...is the actual key:value pair used.

You can use the cell.var comments to find brandable variables and change their values. The comments also help you understand what those variables do. For example, in the given example above, if you were to change the value to “3” (instead of “5”) the Legend would only display three priority options.



The comments are removed at publish-time so they are never visible to users. The comments, also, can be customized.

About brandvars.var

The brandvars.var file is a vartab file that contains definitions to backup the CSS definitions. The brandvars.var file comes with miravar_ variable definitions matching the CSS definitions in the style.css file. You can also use the brandvars.var file to create your own variable definitions using the custvar_ syntax. Keys to be substituted must begin with “custvar_”. For example:

```
“custvar_hostname” = “www.example.com”;
```

Sample usage in an HTML template would be:

```
...<a href="$(custvar_hostname)”>...
```

About locale.var and errstrings.var

The locale.var file contains a string table that Mirapoint **localizes** (translates into different languages). The errstrings.var file contains error messages that can be localized. These vartab files are sometimes referred to as **string tables**. These files contain a list of key:value pairs, where the key is the message name and the value is the localization of the message. The text (key) on the left is replaced with the text (value) on the right during run-time. For example:

```
"Add" = "Add";
```

When localized might be:

```
"Add" = "Ajouter";
```

In locale.var files you can:

- ◆ Modify text within quotes to the right of the equal sign.
- ◆ Put spaces between lines for clarity.

In locale files *do not*:

- ◆ Modify text to the left of the equal sign.
- ◆ Remove the equal signs.
- ◆ Remove the semicolon at the end of the line.

About the Locale Directories

The en_US.ISO_8859-1 directory follows the RFC 1766 naming standard:

- ◆ Two letter lower case language code followed by an underscore (_)
- ◆ Two letter country code in upper case and followed by a period (.)
- ◆ The character encoding



The default locale, en_US.ISO_8859-1, is required. Other locales are optional; they can be set using the CLI command **Locale Set Default**. For branding you can modify the en_US.ISO_8859-1 locale in the existing directory location, or copy the directory hierarchy to a new name of your choosing before modification. For localization you should always keep the RFC 1766 naming convention when copying the directory hierarchy to a new location.

Other Vartab Files

In addition to `cell.var` and `locale.var`, which are unique in each application directory, the different applications have specific vartab files. New vartab files may be added to support new features; other existing vartab files are described in [Table 18](#).

Table 18 Other vartab Files used by the Applications

	File	Description
mail	charsets.var	Character set name definitions for the Message Charsets list used by the Preferences and Compose pages (<code>prefs.html</code> and <code>compose.html</code>). The vartab key is the actual string that gets inserted into the mail message header (e.g., <code>charset="UTF-8"</code>). The vartab values are the localized strings which are displayed in the Message Charset menus.
	timezones.var	Timezone name definitions for the Time Zones list used by the Preferences page (<code>prefs.html</code>). The vartab keys are used by the system to set the users' timezone. Do not modify the keys; you may delete entire entries, if you wish. The vartab values are the localized strings which are displayed in the Time Zones menu.
addrbook	csvheaders.var	Contains the field names for CSV files produced by Outlook. See Localized Import/Export in Address Book on page 87 for details.
user	msgfilterheaders.var	Header definitions for the list displayed in the Message Filter page. The first column is the criteria that is used in the administration protocol. The second column is the unlocalized name that is displayed in the Message Filter combo box. Each name in the second column is looked up in <code>locale.var</code> file to allow for localization.
mcal	mcaldmsg.var	Reminder and invitation format definitions used by Calendar. These messages are documented in "More WebCal Branding" on page 77.

WebMail Standard Edition HTML Template Files Per Mode

Table 19 describes the WebMail HTML template files. **NF** = No Frames mode, **F** = Frames mode, and **F+JS** = Frames with JavaScript mode.

Files that are not described in this document are not linked (underlined).

Table 19 WebMail (mail) HTML Template Files

Template File	NF	F	F+JS	Description
banner.html (WebMail)		✓	✓	Creates the top banner frame in the WebMail pages.
checkmail_s.html	✓	✓		Creates the page that displays the toolbar buttons, the message list column headings, and message list for the Inbox page.
compose.html	✓	✓	✓	Creates the message Compose page that opens when a user clicks Compose .
composetimeout.html	✓	✓	✓	Creates the Mail Session Timed Out page.
confirmattach.html	✓	✓	✓	Confirmation page for attachment requests.
confirmdel.html	✓	✓	✓	Confirmation page for delete requests.
confirmprefs.html/ confirmwindow.html	✓	✓	✓	Confirmation page for preference settings requests.
confirmdel.html	✓	✓	✓	Confirmation page for delete requests.
fetch.html	✓	✓	✓	Creates the URL Fetch page that opens when a user clicks a URL in an open message.
folderedit.html	✓	✓	✓	Creates the Folder Edit page that opens when a user clicks Edit on the Folders page.
folders.html	✓	✓	✓	Creates the Folders list page that opens when a user clicks Folders .
folders_shared.html	✓	✓	✓	Creates the Shared Folders page that opens when a user clicks Shared on the Folders page.
folders_subscribed.html	✓	✓	✓	Creates the Subscribed Folders page that opens when a user clicks Subscribed on the Folders page.
html_compose.html	✓	✓	✓	Creates the HTML-enabled Compose page that opens when a user clicks HTML Text on the Compose page.
links.html (WebMail)		✓	✓	Creates the left frame contents of each page.
login.html	✓	✓	✓	Creates the WebMail Login page that opens when a user accesses the WebMail url, or clicks Logout , or clicks WebMail when in WebCal Direct if both WebMail and WebCal are licensed. This page is fully brandable, you need only retain the system functions and login options.
login_init.html		✓	✓	Creates the WebMail cookies-authentication Login page.
nocookie.html	✓	✓	✓	Cookies must be enabled page.

Table 19 WebMail (mail) HTML Template Files (Continued)

Template File	NF	F	F+ JS	Description
prefs.html	✓	✓	✓	Creates the Preferences page that opens when a user clicks Preferences .
rapidavrescan.html	✓	✓	✓	Currently a blank, place-holder page.
read.html			✓	Creates the Message Contents frame.
read_blank.html			✓	Displays empty contents for the message pane, when no message is selected.
read_s.html	✓	✓		Creates the page that displays your message contents.
readreceipt.html	✓	✓	✓	Creates the Read Receipt Confirmation page that opens when a message is received for which a read receipt has been requested.
refresh.html	✓	✓	✓	Recalls read.html.
retry.html	✓	✓	✓	Creates a “Unable to display your HTML attachment” page with directions.
search_s.html	✓	✓		Creates the page that displays the Searchform, the Toolbar, the Table of Contents column headings, and message list for the Search page.
searchform.html			✓	Creates the Searchform frame.
selcolor.html	✓	✓	✓	Creates the Text Background Color option window for Compose in HTML Text mode.
selfontname.html	✓	✓	✓	Creates the Text Font option window for Compose in HTML Text mode.
selfontsize.html	✓	✓	✓	Creates the Text Font Size option window for Compose in HTML Text mode.
selparagraph.html	✓	✓	✓	Creates the Text Paragraph Tag option window for Compose in HTML Text mode.
spam.html	✓	✓	✓	Creates the This is Spam page that opens when a user clicks the This is Spam button in a message window.
timeout.html	✓	✓	✓	Creates a “Session has timed out” page that opens when there is no user activity for a period of time.
toc.html			✓	Creates the page that displays the Table of Contents column headings for the Inbox , Search , and all Checkmail pages and the message list table.
toolbar.html		✓	✓	Creates the Toolbar frame for the Inbox , Search , and all Checkmail pages.
urlid_timeout.html	✓	✓	✓	Creates the Link Timed Out page that opens when a user clicks a non-active link in a message.

About the HTML Template Files

The template files contain HTML code, as well as variables that are replaced by the system with appropriate values; and all of the files contain a reference to the CSS fallback variables in `brandvars.var`. These template files end with the suffix “.html.”

When customizing the template files you must keep in mind that nearly all of the files have dependencies on other files. For example, in order to change the order of the data columns in WebMail, you would need to change the order of the table data variables defined in `cell.var` as well as change the order of the table column headers as positioned in the template file.

Often you may want to change a screen element and can use the comments in the `cell.var` or the template file to help.

Once you make your changes, you must zip the file set and publish it in order to view your results.



Stop and re-start WebMail and WebCal for the publishing process to complete.

Reading the Template File Comments

Most of the HTML template files are commented to make it easier for you to find those elements that you can brand and, particularly, to help you find their corresponding `vartab` file key (where applicable).

The following is an example:

```
<!-- @
var: composejs
desc: Javascript used in javascript mode
see also key: composejs
@ -->
```

The comments are usually placed right above the actual corresponding code and can be translated as follows:

“var:” is the variable name.

“desc:” is a description of the variable.

“see also key:” is the key name you can find in the `vartab` file, usually `cell.var`.

Sometimes the key name matches the variable name, sometimes it doesn't.

You can use the template comments to find brandable variables in the `cell.var` file and change their values. The comments also help you understand what those variables do.

More WebMail Branding

This section describes some WebMail branding options that are available to you in addition to the CSS.

Setting a Default Charset for WebMail

To set a default charset definition for an email message that does not have a valid MIME header in it that dictates its charset, the recommendation is the following, as example:

1. Edit file `mail/locale/localename/cell.var` to have this entry:

```
msg_delivery_charset = "ISO-8859-1";
```

This will pick one of the keys from `mail charsets.var` file as the default **Message Charset**.

2. Add this entry to file `mail charsets.var`:

```
"ISO-8859-1" = "Western European (ISO-8859-1)";
```



The right-hand side is only used for display in the **Message Charset** combo box in **Compose**. The key (on the left) is the string that must match the right-hand-side of step #1, and be a supported charset in the libiconv library.

Branding Default Mode Preferences for WebMail

The factory brand comes with the default **Mode** set to **Frames**, unless otherwise requested during provisioning. To brand a default **Mode** of **No Frames** for WebMail, add this line to `mail/locale/localename/cell.var`:

```
versiondefault = "v_noframe";
```

The general rule is to append "default" to the key, and WebMail uses that value (if it's valid) for the initial preferences setting. For example:

```
versiondefault = {v_noframe | v_all | v_nojs};
usetrashdefault = {ut_no | ut_yes | NULL};
savesentdefault = {ss_no | ss_yes | NULL};
includesigdefault = {is_no | is_yes | NULL};
replyoptdefault = {fwd_no | fwd_inline | fwd_attach};
usehtmldefault = {html_no | html_yes | NULL};
```

Table 20 Default Preference Keys Explained

Key and Options		Description
versiondefault	v_noframe	No Frames
	v_all	Frames with JavaScript
	v_nojs	Frames No JavaScript
usetrashdefault	ut_no	Delete to Trash is OFF
	ut_yes	Delete to Trash is ON

Table 20 Default Preference Keys Explained (Continued)

Key and Options		Description
savesentdefault	ss_no	A copy of sent mail is saved
	ss_yes	A copy of sent mail is not saved
includesigdefault	is_no	Personal signature is included
	is_yes	Personal signature is not included
replyoptdefault	fwd_no	The original message is not included in reply messages
	fwd_inline	The original message is included “inline” with the reply messages
	fwd_attach	The original message is included as an attachment
usehtmldefault	html_no	Compositions open in Plain Text mode
	html_yes	Compositions open in HTML mode



The `savesentdefault` and `usetrashdefault` preferences are defined in the system brand `zip mail/.../cell.var`, but the other default preferences available to you (described in [Table 20](#)) must be added to that file. They can be added in the same place.

Additional keys that you may add to set defaults are the shown in the following examples; use an appropriate alphanumeric string as the value. Examples:

```
fullnamedefault = "Unspecified";
emaildefault = "Unspecified";
reply-todefault = "Unspecified";
headerdefault = "Unspecified"
sentfolderdefault = "Sent";
composewidthdefault = "50";
composeheightdefault = "12";
timezonedefault = "GMT+10:00 Australia|Brisbane";
charsetdefault = "ISO-2022-JP";
draftfolderdefault = "Draft";
trashfolderdefault = "Trash";
quota_adjust_storage = "true"
index_msglistnum = "25"
newarrivalfirstdefault = "newarrivalfirst_no"
```

Changing the Date Format in WebMail

In `mail/locale/localename/cell.var` the variable `tocdateformat` contains a `strftime` date format string. The default value is

```
"%b %d %H:%M";
```

To add Year, simply add `%Y` to the string.:

```
"%b %d %H:%M":%Y;
```

Many of the ANSI C function `strftime` format directives may be added here. Each conversion specification is replaced by the characters as described in [Table 21](#).

Table 21 ANSI C Function `strftime` Format Directives

	Is replaced by the...
<code>%A</code>	Locale's full weekday name.
<code>%a</code>	Locale's abbreviated weekday name.
<code>%B</code>	Locale's full month name.
<code>%b</code>	Locale's abbreviated month name.
<code>%c</code>	Locale's appropriate date and time representation.
<code>%d</code>	Day of the month as a decimal number (01-31).
<code>%H</code>	Hour (24-hour clock) as a decimal number (00-23).
<code>%I</code>	Hour (12-hour clock) as a decimal number (01-12).
<code>%j</code>	Day of the year as a decimal number (001-366).
<code>%M</code>	Minute as a decimal number (00-59).
<code>%m</code>	Month as a decimal number (01-12).
<code>%p</code>	Locale's equivalent of either "AM" or "PM".
<code>%S</code>	Second as a decimal number (00-60).
<code>%U</code>	Week number of the year (Sunday as the first day of the week) as a decimal number (00-53).
<code>%W</code>	Week number of the year (Monday as the first day of the week) as a decimal number (00-53).
<code>%w</code>	Weekday (Sunday as the first day of the week) as a decimal number (0-6).
<code>%X</code>	Locale's appropriate time representation.
<code>%x</code>	Locale's appropriate date representation.
<code>%Y</code>	Year with century as a decimal number.
<code>%y</code>	Year without century as a decimal number (00-99).
<code>%Z</code>	Time zone name.
<code>%%</code>	"%".

Changing How WebMail Handles Attachments

You can change WebMail to handle all attachments the way it handles .doc, .pdf and many other file types: just display an **Open** link to the file but do not display the file inline. To do this, change these lines to your `mail/locale/localename/cell.var` to surround the inline attachments in HTML comments:

```
msg_start_tag = "<pre>";
msg_end_tag = "</pre>";
```

To these:

```
msg_end_tag = "// -->";
msg_start_tag = "<!--";
```

To cover text attachments that have any lines exceeding 82 bytes, you'll need to change these lines:

```
long_msg_start_tag = "<tt>";
long_msg_end_tag = "</tt>";
```

To these:

```
long_msg_end_tag = "// -->";
long_msg_start_tag = "<!--";
```

To cause image files not to display inline, replace this line:

```
printinline = "<$(tag) src=\"$(HTTP)://$(HOST)/wm/mail/
genimage$(ext)?sessionid=$(sessionid)&uid=$(uid)&off=$(off)&len=$(len)&enc
=$(enc)&typ=$(type)\" alt=\"\" border=0$(other)>";
```

With this:

```
printinline = "";
```

Branding the WebCal Direct Pages

This section provides an overview of each file in the `mcal/locale/localename/` application directory that defines the pages for WebCal Direct (personal calendar) and Groupcal Direct (group calendar) in Standard Edition. Groupcal is a super-set of WebCal, so you must have both licenses in order to get the Groupcal functions. Described are the changeable elements, variable values and HTML tags.



Many provided variables, hidden input types, and code should not change because it can break your application.

About the WebCal Pages

The files that create the WebCal pages are located in the `mcal/locale/localename/` directory. Like WebMail, WebCal has its own set of frame files. The pages are created by a combination of process files; for example `cell.var`, and HTML template files that are supplied with data by the system. The HTML template files reference a CSS (cascading style sheet). The style elements defined in the CSS are also defined in a special process file, `brandvars.var`, so browsers that do not support CSS still display specified style formatting.



Described are the files for Groupcal Direct, in which WebCal Direct is a subset. Where elements apply to both and are defined by variables, two screenshots are shown.



Default Calendar permissions allow Calendar to do free/busy lookups on all Calendar users in the same domain, including cross-box domains. In that manner, users can see when other Calendar users are available. For some other Calendar options, such as viewing or subscribing to another's Calendar, specific permissions must be set; where these apply a note is made.

WebCal Addressbook

WebCal Address Book uses the same files WebMail Address Book uses. Dollar sign (\$) variables are used to call out the differences between the two applications.

About the WebCal Viewing Modes

WebCal provides users with two different viewing mode options. The **Frames with JavaScript** mode in WebCal only effects WebMail; if WebMail is not also licensed, this option is not available.

Frames Mode

[Figure 22](#) on page 72 is an example of the WebCal **Preferences** page **User Information** view in the **Frames** mode. A frame definition file, `window.html`, defines a frameset that references the **banner** (`banner.html`), **links** (`links.html` and `blank.html`) and the HTML template file, `prefs_gen.html` if Groupcal is licensed; `prefs.html` if only WebCal is licensed.

When you click on a **Preferences** view link, that file (permissions.html or subscriptions.html) replaces prefs_gen.html. [Figure 22](#) on page 72 shows what the Calendar Preferences page **User Information** view in **Frames** mode looks like.

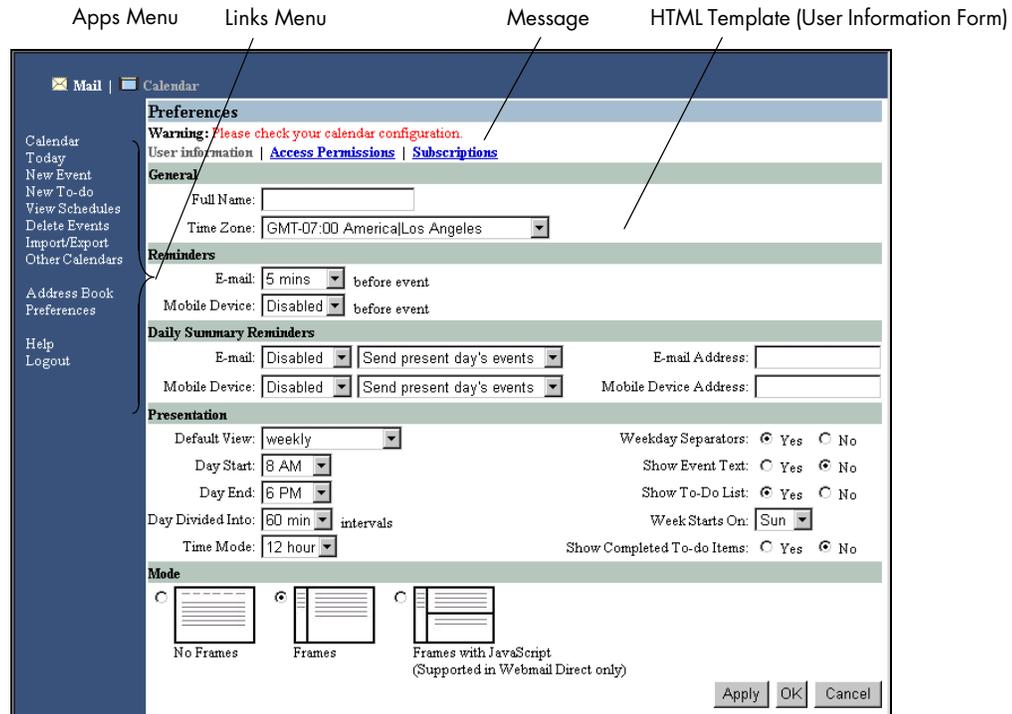


Figure 22 WebCal Screen Elements—Frames Mode

No Frames Mode

[Figure 23](#) is an example of the WebCal Preferences page **User Information** view in the **No Frames** mode.

When you click on a **Preferences** view options link (only available for Groupcal), that file (permissions.html or subscriptions.html) opens and prefs_gen.html goes away. [Figure 23](#) shows what the Calendar Preferences page **User Information** view in **No Frames** mode looks like.

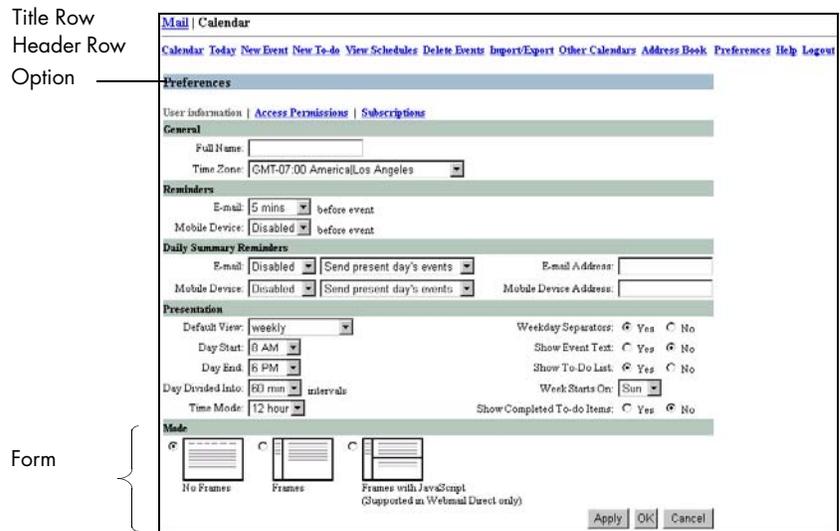


Figure 23 WebCal Screen Elements—No Frames Mode

About the WebCal Screen Elements

Throughout WebCal these elements display in a frame file such as banner.html or links.html; variables tell the system how to display the:

- ◆ **Apps Menu:** Only displays if WebCal Direct and WebMail Direct are both licensed. Active link changes depending on which application you are in.
- ◆ **Links Menu or Simplelinks:** Defined in links.html in a frame mode. Or as a variable \$(simplelinks) in the HTML template files for **No Frames** mode.

How the WebCal Pages are Defined

This section describes the frame, frameset, process (vartab and so forth), and HTML template files that define the pages. Some CSS selectors defined in style.css are used on every page. Refer to [Working with the Cascading Style Sheets](#) on page 20 for details on the CSS.

WebCal Frameset Definition Files

Table 22 describes the WebCal frame definition files. These files are only used in the **Frames** and **Frames with JavaScript** modes. You may need to adjust the frame set definitions in these files if you reorder elements or add graphics.

Table 22 WebCal Frameset Definition Files

	Description & Dependencies
blank.html	A frame definition file that ensures that the correct files display in the correct order.
window.html	Controls which files are displayed and how much room they take up. References banner.html, links.html and blank.html

WebCal Process Files

Table 23 describes the WebCal process files that are used to process JavaScript and system requests. You can make global changes by changing variable definitions in these files.

Table 23 WebCal Process (mcal/) Files

	Description & Dependencies
cell.var	Contains variable definitions.
locale.var	Contains customizable system and error messages.
mcaldmsg.var	Contains customizable Calendar messages.
timezones.var	Contains timezone name definitions.
cal.js	Contains the JavaScript that the Select , Clear All , and Find functions use in the Frames with JavaScript mode.

WebCal HTML Template Files

Table 24 describes the template files. You can change element terminology, colors (in some cases), add graphics, and reorder data fields using these files. These files are used by all three modes. In the **No Frames** mode, system variables tell the system to display elements that the frames and modes do not display because they use frame files instead.

Files that are not described in this document are not linked (underlined).

Table 24 WebCal (mcal/) HTML Template Files

Template File	Description
attendee.html	The Accept/Decline page for invited attendees.
banner.html (WebCal)	Creates the Banner frame at the top of each page.
confirmprefs.html	Confirmation page for preferences requests.

Table 24 WebCal (mcal/) HTML Template Files (Continued)

Template File	Description
daily.html	Creates the Calendar page in the Daily View that opens when a user clicks the Daily View icon in the thumbnail calendar on the Calendar page. Also opens for the current day when a user clicks the Today link; or for a selected day when a user clicks a Day link in the thumbnail calendar.
delete.html	Creates the Delete Events page that opens when a user clicks Delete Events .
deleteconfirm.html	Confirmation page for delete requests.
download.generic.html	Allows you to brand the default Outlook Synq download.html file.
errorpage.html	Creates the Server Error page.
eventro.html	Creates the Event profile page for a read-only event.
extview.html	Creates the Accept/Decline Event page for invited external attendees.
extviewconfirm.html	Event acceptance confirmation page for external attendees.
extviewerror.html	Error message page for the Accept/Decline Event page for external attendees.
links.html (WebCal)	Creates the left frame contents of each page.
login.html	Creates the WebCal Login page that opens when a user accesses the WebCal url, or clicks Logout . This page is fully brandable, you need only retain the login options.
login_init.html	Creates the WebCal cookies-authentication Login page.
monthly.html	Creates the Calendar page in the Monthly View that opens when a user clicks the Monthly View icon in the thumbnail calendar on the Calendar page.
nav.html	Creates the date calendar on the left hand side.
new.html	Creates the Add/Edit Event page that opens when a user clicks New Event or the Title link for an exiting event. Allows users to configure events or meetings; a meeting is an event to which other people are invited.
newtodo.html	Creates the Add/Edit To-do page that opens when a user clicks New To-do or the Title link for an exiting to-do item. Allows users to configure to-do items.
nocookie.html	Cookies must be enabled page.
permissions.html	Creates the Groupcal Direct Preferences page in the Access Permissions view that opens when a user clicks the Access Permissions link on one of the Preferences page views.

Table 24 WebCal (mcal/) HTML Template Files (Continued)

Template File	Description
pick_daily.html	Creates the View Schedules page that opens when a user clicks View Schedules . Allows users to view the dates and times of other Calendar users.
prefs.html	Creates the Preferences page when only WebCal Direct (personal calendar) is licensed.
prefs_gen.html	Creates the Groupcal Direct Preferences page in the User Information view that opens when a user clicks the Preferences link or the User Information link on one of the other Preferences page views.
prefs_mtg.html	Creates the Preferences Meeting Options page. Reserved for use in a future release.
prn_dyvw.html	Creates the Calendar Daily Print Preview page.
prn_mnvw.html	Creates the Calendar Monthly Print Preview page.
prn_whvw.html	Creates the Calendar Weekly horizontal Print Preview page.
prn_wkvw.html	Creates the Calendar Weekly vertical Print Preview page.
repeat_view.html	Creates the Add/Edit Event page List all Instances and List all Exceptions views for repeating events.
subscriptions.html	Creates the Groupcal Direct Preferences page in the Subscriptions view that opens when a user clicks the Subscriptions link on one of the Preferences page views.
sync.html	Creates the Import/Export page that opens when a user clicks Import/Export .
timeout.html	Creates a “Session has timed out” page.
today.html	Creates a distilled version of the Daily View page for the current day. This file is not currently supported for general use.
userslocaltime.html	Displays the current time. This file is not currently supported for general use.
viewother.html	Creates the Other Calendars page that opens when a user clicks Other Calendars . Allows users to view the Calendars of other users on the system. Special permissions are required.
viewotherdenied.html	Error message for the View Others page when the necessary permissions are not set.
weekly.html	Creates the Calendar page in the Weekly View (the default vertical mode). This page is usually the default page that opens when a user clicks Calendar or clicks the Weekly Vertical icon in the thumbnail calendar on the Calendar page.
weeklyhoriz.html	Creates the Calendar page in the Weekly View horizontal mode that opens when a user clicks the Weekly Horizontal icon in the thumbnail calendar on the Calendar page.

More WebCal Branding

This section describes some of additional WebCal branding options that are available to you.

Supporting Group Calendar

You must add the following to your `mcal/locale/localename/cell.var` file (comments included) in order to support group calendar:

```
# key: group_calendar_brand
# desc: This variable should be set to 'yes' if this brand
# contains group calendar support
group_calendar_brand="yes";
```

Branding Default Mode Preferences for WebCal

The WebCal preferences defaults are set in the `mcal/locale/localename/cell.var` file. Unlike WebMail, rather than appending “default” to the key; you change the value. In addition, there are many more display factors and options that can be set in this file ([Table 25](#)).

Table 25 WebCal Default Preference Keys Explained

Key and Options		Description
force_init_config	yes	Forces users to set their preferences on login if they haven't yet
	no	
weekstart	0 - 6	Default day week starts on for users. 0=Sunday, 1=Monday, 2=Tuesday, and so forth.
default_time_mode	12	Determines if times are shown in 12 or 24 hour mode.
	24	

Branding Calendar Email Messages

Calendar has many different emails that it sends out for different circumstances. Email messages are defined in `mcal/locale/localename/mcaldmsg.var`, some email variables are defined in `mcal/locale/localename/cell.var`.

One new variable supports the branding of the accept/decline URL in the event invitation message:

```
#key: extview_hostname
#desc: Override the hostname used in the accept/decline URLs
# sent via email to external attendees. If this variable
# is empty, the hostname used in the URL is the configured
# hostname of the mailstore (see "Dns Set domain" in the CLI).
extview_hostname = "";
```

The different email message types and their variables are described in [Table 26](#). In addition, see `mcaldmsg.var` for other variable definitions.

Table 26 Email Messages Defined in `mcaldmsg.var`

Variable	Default Message	Description
<code>invitationnew</code>	<p>“You are invited to attend the following meeting:</p> <p>Title: <code>\$(synopsis)</code> Owner: <code>\$(owner)</code> Date: <code>\$(eventdate)</code> Time: <code>\$(eventtime)</code> Participants: <code>\$(attendees)</code> <code>\$(location_section)</code> Description: <code>\$(description)”</code></p>	<ul style="list-style-type: none"> ❖ Sent to Meeting Attendees when a new meeting is created and Send E-mail is selected ❖ These variables are defined by user input
<code>external_invitationnew</code>	<p>“You are invited to attend the following meeting:</p> <p>Title: <code>\$(synopsis)</code> Owner: <code>\$(owner)</code> Date: <code>\$(eventdate)</code> Time: <code>\$(eventtime)</code></p> <p>To confirm or decline attendance, please visit the following web page:</p> <p><code>\$(externalurl)</code> <code>\$(location_section)</code> Description: <code>\$(description)”</code></p>	<ul style="list-style-type: none"> ❖ Sent to External Meeting Attendees when a new meeting is created and Send E-mail is selected ❖ These variables are defined by user input

Table 26 Email Messages Defined in mcaldmsg.var

Variable	Default Message	Description
invitationsubject	<p>“\$(owner): \$(synopsis)”; external_invitation= “The following meeting has been modified:</p> <p>Title: \$(synopsis) Owner: \$(owner) Date: \$(eventdate) Time: \$(eventtime)</p> <p>To confirm or decline attendance, please visit the following web page:</p> <p>\$(externalurl) \$(location_section) Description: \$(description)”</p>	<ul style="list-style-type: none"> ❖ Sent to Meeting Attendees when a meeting is modified by the meeting owner ❖ These variables are defined by user input
external_invitationsubject	<p>“\$(owner): \$(synopsis)”; external_invitation= “The following meeting has been modified:</p> <p>Title: \$(synopsis) Owner: \$(owner) Date: \$(eventdate) Time: \$(eventtime)</p> <p>To confirm or decline attendance, please visit the following web page:</p> <p>\$(externalurl) \$(location_section) Description: \$(description)”</p>	<ul style="list-style-type: none"> ❖ Sent to External Meeting Attendees when a meeting is modified by the meeting owner ❖ These variables are defined by user input

Table 26 Email Messages Defined in mcaldmsg.var

Variable	Default Message	Description
acceptsubject	<p>“meeting accepted by \$(attendee)”;</p> <p>accept =</p> <p>“\$(attendee) has accepted your invitation to:</p> <p>Title: \$(synopsis)</p> <p>Owner: \$(owner)</p> <p>Date: \$(eventdate)</p> <p>Time: \$(eventtime)</p> <p>Participants: \$(attendees)”</p>	<ul style="list-style-type: none"> ❖ Sent to the meeting owner when Accept is clicked by an invited meeting attendee ❖ These variables are defined by user input
declinesubject	<p>“meeting declined by \$(attendee)”;</p> <p>decline =</p> <p>“\$(attendee) has declined your invitation to:</p> <p>Title: \$(synopsis)</p> <p>Owner: \$(owner)</p> <p>Date: \$(eventdate)</p> <p>Time: \$(eventtime)</p> <p>Participants: \$(attendees)”</p>	<ul style="list-style-type: none"> ❖ Sent to the meeting owner when Decline is clicked by an invited meeting attendee ❖ These variables are defined by user input
cancellationsubject	<p>“\$(owner): \$(synopsis)”;</p> <p>cancellation=</p> <p>“This meeting has been cancelled by the owner:</p> <p>Title: \$(synopsis)</p> <p>Owner: \$(owner)</p> <p>Date: \$(eventdate)</p> <p>Time: \$(eventtime)</p> <p>Description:</p> <p>\$(description)”</p>	<ul style="list-style-type: none"> ❖ Sent to Meeting Attendees when a new meeting is created and then canceled by the meeting owner ❖ These variables are defined by user input
external_cancellationsubject	<p>“\$(owner): \$(synopsis)”;</p> <p>external_cancellation=</p> <p>“This meeting has been cancelled by the owner:</p> <p>Title: \$(synopsis)</p> <p>Owner: \$(owner)</p> <p>Date: \$(eventdate)</p> <p>Time: \$(eventtime)</p> <p>Description:</p> <p>\$(description)”</p>	<ul style="list-style-type: none"> ❖ Sent to External Meeting Attendees when a new meeting is created and then canceled by the meeting owner ❖ These variables are defined by user input

Table 26 Email Messages Defined in mcaldmsg.var

Variable	Default Message	Description
dailysummary	“Summary of events for \$(date): Time Title ---- ---- \$(eventlist)”	<ul style="list-style-type: none"> ❖ Sent to the calendar owner when Daily Summary Reminders for the present day is selected on the Preferences page User Information view ❖ These variables are defined by user input
dailysummarytomorrow	“Summary of events for \$(tomorrow): Time Title ---- ---- \$(eventlist)”	<ul style="list-style-type: none"> ❖ Sent to the calendar owner when Daily Summary Reminders for the following day is selected on the Preferences page User Information view ❖ These variables are defined by user input

Branding the Options/Account Pages

This chapter provides an overview of the files in the **user** directory that define the **Options** pages for WebMail and the **Account** pages for the Administration Suite.



When the user is in **Frames** or **Frames with JavaScript** mode, the **Options** pages display in a frame along with the WebMail Banner (banner.html) and Links (links.html) frames.



There are many variables, hidden input types, and code that you should not change. Doing so may break your application. This section describes the elements, and variable values that you can change and offers guidelines for HTML changes that you can make.

About the Options/Account Pages

The files that create the **Options/Account** pages are shared by WebMail (**Options**) and the Administration Suite (**Account**) and are located in the `user/locale/localename/` directory. The pages are created by a combination of process files that manage requests, and HTML files that provide templates for the display. The HTML template files reference a CSS (cascading style sheet). The style elements defined in the CSS are also defined in a special process file, `brandvars.var`, so browsers that do not support CSS still display specified style formatting.

Options/Account Process Files

These files are used to process JavaScript and system requests. You can make global changes by changing variable definitions in these files.

Table 27 Options/Account (user/) Process Files

File Name	Description and Dependencies
cell.var	Contains variable definitions.

Table 27 Options/Account (user/) Process Files (Continued)

File Name	Description and Dependencies
locale.var	Contains customizable system and error messages.
msgfilterheaders.var	The list of headers displayed in the Message Filter page.

Options/Account HTML Template Files

Table 28 describes the template files. You can change element terminology, colors (in some cases), add graphics, and reorder data fields using these files. These files are used by all three modes. In the **No Frames** mode, system variables tell the system to display elements that the **Frames** and **Frames with JavaScript** modes do not display because they use frame files instead.

Files that are not described in this document are not linked (underlined).

Table 28 Options/Account (user/) HTML Template Files

File Name	Description
antispamfilter.html	Creates the Junk Mail Control page. The Junk Mail Filter view page that opens when a user clicks Junk Mail Control and then the Junk Mail Filter link in the Options (WebMail) or Account (Suite) pages.
autoreply.html	Creates the Auto Reply page that opens when a user clicks Automatic Reply in the Options (WebMail) or Account (Suite) pages.
blacklist.html	Creates the Junk Mail Control page Blocked Senders view that opens when a user clicks Junk Mail Control in the Options (WebMail) or Account (Suite) pages.
chgpas.html	Creates the Change Password page that opens when a user clicks Change Password in the Options (WebMail) or Account (Suite) pages.
deletegetmail.html	Creates the Delete POP Server page that opens when a user clicks the Delete icon in the External Mail WebMail page.
editgetmail.html	Creates the Add/Edit POP Server page that opens when a user clicks Add POP Server or the Edit icon in the External Mail WebMail page.
filter.html	Creates the Message Filters page that opens when a user clicks Message Filters in the Options (WebMail) or (Suite) Account pages.
forward.html	Creates the Forwarding page that opens when a user clicks Forwarding in the Options (WebMail) or Account (Suite) pages.
getmail.html	Creates the External Mail page that opens when a user clicks External Mail in WebMail.
login.html	Used by the Administration Suite.

Table 28 Options/Account (user/) HTML Template Files

File Name	Description
setacl.html	Creates the Access Control page. that opens when a user clicks Options in WebMail, Account in the Suite, or Access Control in either. This page is defined by the Account Administration frame, and the banner and links frames when Frames are enabled.
whitelist.html	Creates the Junk Mail Control page Allowed Senders view that opens when a user clicks Junk Mail Control in the Options (WebMail) or Account (Suite) pages.
whitelistto.html	Creates the Junk Mail Control page Allowed Mailing Lists view that opens when a user clicks Junk Mail Control in the Options (WebMail) or Account (Suite) pages.

Branding the Spelling Pages

This chapter provides an overview of the files in the `spell/locale/localename/` directory that define the Spelling pages for WebMail.



When the user is in **Frames** or **Frames with JavaScript** mode, the Spelling pages display in a frame along with the WebMail Banner (`banner.html`) and Links (`links.html`) frames.

About the Spelling Pages

The Spelling pages are used by WebMail and located in a separate directory: `spell/locale/localename/`. The pages are created by a combination of process files that manage requests, and HTML files that provide templates for the display. The HTML template files reference a CSS (cascading style sheet). The style elements defined in the CSS are also defined in a special process file, `brandvars.var`, so browsers that do not support CSS still display specified style formatting.

Spelling Process Files

These files are used to process system requests. You can make global changes by changing variable definitions in these files.

Table 29 Spell (spell/) Process Files

File Name	Mode(s)	Description
cell.var	All modes	Contains variable definitions.
locale.var	All modes	Contains customizable system and error messages.

Spelling HTML Files

These are the template files. You can change element terminology, colors (in some cases), add graphics, and reorder data fields using these files. These files are used by all three modes. In the **No Frames** mode, system variables tell the system to display

elements that the **Frames** and **Frames with JavaScript** modes do not display because they use frame files instead.

Table 30 Spelling (spell/) HTML Template Files

File Name	Mode(s)	Description
editdictionary.html	All modes	Creates the Spelling: Dictionary page that opens when a user clicks Edit when checking spelling in the Compose window.
spell.html	All modes	Creates the Spelling page that opens when a user clicks Spelling when in the Compose window.
timeout.html	All modes	Creates a “Session has timed out” page.

Branding the Address Book Pages

This chapter provides an overview of the files in the `addrbook/locale/localename/` directory that define the Address Book pages for WebMail and WebCal

WebMail Address Book uses the same files WebCal Address Book uses though the pages displayed are slightly different. Address Book elements are displayed by the use of variables that turn on/off WebMail vs. WebCal elements.



When the user is in **Frames** or **Frames with JavaScript** mode, the Address Book pages display in a frame along with the WebMail Banner (`banner.html`) and Links (`links.html`) frames.



There are many variables, hidden input types, and code that you should not change. Doing so may break your application. This section describes the elements, and variable values that you can change and offers guidelines for HTML changes that you can make.

About the Address Book Pages

The files that create the Address Book pages are shared by WebMail and WebCal and are located in the `addrbook/locale/localename/` directory. The pages are created by a combination of process files that manage requests, and HTML files that provide templates for the display. The HTML template files reference a CSS (cascading style sheet). The style elements defined in the CSS are also defined in a special process file, `brandvars.var`, so browsers that do not support CSS still display specified style formatting.

Address Book—WebMail vs. WebCal

Address Book for WebMail Direct is slightly different than Address Book for WebCal Direct. Variables are used to change elements when Address Book is accessed by WebMail or by WebCal. In particular, the Simplelinks variable changes to display a different link set for WebMail Address Book vs. WebCal Address Book. [Figure 24](#) shows the Address Book Simplelinks menu for WebMail; [Figure 25](#) shows the Address Book Simplelinks menu for WebCal.

[Check Mail](#) [Inbox](#) [Compose](#) [Folders](#) [Search](#) [Address Book](#) [Preferences](#) [Options](#) [Compact](#) [External Mail](#) [Help](#) [Logout](#)

Figure 24 Address Book Simplelinks Menu for WebMail Direct

[Calendar](#) [Today](#) [New Event](#) [To-do Item](#) [Delete Events](#) [Import/Export](#) [Address Book](#) [Preferences](#) [Help](#) [Logout](#)

Figure 25 Address Book Simplelinks Menu for WebCal Direct

Address Book Process Files

These files are used to process JavaScript and system requests. You can make global changes by changing variable definitions in these files.

Table 31 Address Book (addrbook/) Process Files

	Description and Dependencies
cell.var	Contains variable definitions.
csvheaders.var	Contains the field names for CSV files produced by Outlook. See Localized Import/Export in Address Book on page 87 for details.
locale.var	Contains customizable system and error messages.

Address Book HTML Template Files

[Table 32](#) describes the template files. You can change element terminology, colors (in some cases), add graphics, and reorder data fields using these files. These files are used by all three modes. In the **No Frames** mode, system variables tell the system to display elements that the frames and modes do not display because they use frame files instead.

Table 32 Address Book (addrbook/) HTML Template Files

File Name	Description
categories.html	Creates the Edit Categories page that opens when a user clicks Edit Categories on the Contacts page.
confirmdelete.html	Creates the Confirm Delete page that opens when a user clicks Delete on the Contacts , Groups , Directory Services , or Edit Categories pages.
conflict.html	Creates the Duplicate Contact page that opens when a user imports an address book file that has entries already in Address Book.
contactprofile.html	Creates the Contact Profile page that opens when a user clicks a Name link on the Contacts page or in the search results (Find People) performed on a non-LDAP directory server.
contacts.html	Creates the Contacts page that opens when a user clicks Address Book , or Contacts .
currentldapselect.html	Creates the Import to Address Book page that opens when a user selects the Import check box for a Name link in a LDAP directory server search result and clicks Import .

Table 32 Address Book (addrbook/) HTML Template Files (Continued)

File Name	Description
currentselect.html	For WebMail, creates the E-mail To/Cc/Bcc : contacts page that opens after a user has made selections in Address Book and clicked Compose . For WebCal and Groupcal, creates the Event Notification To : contacts page that opens after a user has made selections in Address Book and clicked New Event .
editcontact.html	Creates the Add/Edit Contact page that opens when a user clicks Add Contact or the Edit icon on the Contacts page, or the Edit button on the Contact Profile page.
editgroup.html	Creates the Add/Edit Group page that opens when a user clicks Add Group or the Edit icon on the Groups page, or the Edit button on the Group Profile page.
editldap.html	Creates the Directory Services Properties page that opens when a user clicks Add Service in the Directory Services page, or highlights a service and clicks Properties in the Directory Services page.
groupprofile.html	Creates the Group Profile page that opens when a user clicks a Name link on the Groups page.
groups.html	Creates the Groups page that opens when a user clicks Groups .
impexp.html	Creates the Import/Export page that opens when a user clicks Import/Export .
ldapctprop.html	Creates the Contact Profile page that opens when a user clicks a Name link in the search results (Find People) performed on a LDAP directory server.
ldapdirs.html	Creates the Directory Services page that opens when a user clicks Directory Services on the Find People page.
outofdate.html	Creates the Calendar Out of Date page that opens when a user's Address Book was changed by another session, or by an administration command and the current session needs to re-synchronize.
search.html	Creates the Find Contacts page that opens when a user clicks Find Contacts .
searchgroup.html	Creates the Find Groups page that opens when a user clicks Find Groups on the Groups page.
timeout.html	Creates the Session has Timedout page that opens when there is no user activity for a period of time.

More Address Book Branding

This section describes some of the additional Address Book branding options that are available to you.

Localized Import/Export in Address Book

In order to support the ability to specify the character set of a file used to import data into address book, the vartab file `addrbook/locale/localename/csvheaders.var` provides the field name definitions for CSV files produced by Outlook. A record in this file is composed of up to six parameters, describing Outlook CSV fields for a given language:

- ◆ *language_label*: This entry contains the text that is displayed for this language in the drop list on the **Import/Export** page where the user can select the CSV file language.
Example: `english_label="English";`
- ◆ *language_enabled*: This optional entry helps control whether a given record is enabled or not. It defaults to **true**; if **false**, the given language is not available on the **Import/Export** page.
Example: `english_enabled=true;`
- ◆ *language_default*: This optional entry indicates which record is selected in the list when the **Import/Export** page is first displayed. It defaults to **false**; if **true**, the given language is the default selection when importing or exporting the address book in CSV format.
Example: `english_default=true;`
- ◆ *language_default_charset*: This entry contains the character set that is usually used for this language. For example, for Japanese, it would be **Shift_JIS**. This value is used for import and export operations when the selected character set is **default**.
Example: `english_default_charset="UTF-8";`
- ◆ *language_charset*: This entry contains the character set of the localized data in the vector that is the value of the last entry, *language_contents*. This entry is optional and defaults to **UTF-8**.
Example: `english_charset="UTF-8";`
- ◆ *language_contents*: This entry contains a vector of an even number of strings. These are the CSV field names, exactly as they appear in a CSV file produced by Outlook (or Outlook Express) for the corresponding language. The vector contains, in odd-indexed positions, the English field names, and, in even-indexed, positions the localized field names. Only the second set of strings may be converted using the previous *language_charset* entry. It is recommended that all values be UTF-8.
Example:

```
english_contents=(
  "Title",           "Title",
  "First Name",     "First Name",
  "Middle Name",    "Middle Name",
  "Last Name",      "Last Name",
  "Suffix",         "Suffix",
  "Company",        "Company",
```

```

"Department",
"Job Title",
"Business Street",
"Business Street 2",
"Business Street 3",
"Business City",
"Business State",
"Business Postal Code",
"Business Country",

... more fields ...
);

```

The default `csvheaders.var` comes with some languages (English, French, Italian, Spanish, Portuguese, Chinese, Taiwanese, Korean, German). A disabled entry, `english_short`, lists the fields that are supported by Address Book. You may add your own language, and control which languages are enabled.

Branding Contact Data

The Address Book pages **Contacts** (`contacts.html`), **Contact Profile** (`contactprofile.html`), **Find People** (`search.html`), **Confirm Contact Delete** (`confirmdelete.html`), and **Add/Edit Contact** (`editcontact.html`), contain table data created by the variables described in [Table 33](#). **Note:** “mcal” indicates a variable used by Calendar Address Book.

Table 33 Contact Table Row Format Variables

Variable	Description
<code>contact_tr</code>	One row in the list of contacts
<code>mcal_contact_tr</code>	
<code>delete_1ct_tr</code>	One row in the list of contacts when deleting one contact
<code>delete_ct_tr</code>	One row in the list of contacts when deleting several contacts
<code>searchresult_tr</code>	One row in the list of contacts for one contact found in the address book
<code>mcal_searchresult_tr</code>	
<code>ldap_searchresult_tr</code>	One row in the list of contacts for one contact found in LDAP
<code>mcal_ldap_searchresult_tr</code>	

The fields of the current contact are defined through the following variables described in [Table 34](#).

Table 34 Contact Table Row Data Variables

Variable	Description
<code>categoryIndex</code>	The index of the category to which this contact belongs.
<code>categoryName</code>	The localized name of the category to which this contact belongs.
<code>primarynumber</code>	The primary phone number for the contact.

Table 34 Contact Table Row Data Variables (Continued)

Variable	Description
lastname	Contact's last name.
firstname	Contact's first name.
email	Contact's email address.
email_link mcal_email_link	The contact's email address as a link to WebMail or Calendar.
company	The contact's company.
department	The contact's department.
title	Contact's job title
note	Contact's description
street	Contact's street address
city	Contact's city
state	Contact's state
zip	Contact's zip code
country	Contact's country code
workphone	Contact's work phone number
homephone	Contact's home phone number
cellphone	Contact's cell phone number
pager	Contact's pager number
fax	Contact's fax number
nickname	Contact's nick name
url	Contact's company URL or personal web page
displayname	Contact's display name
ad_y	Contact's anniversary year
ad_m	Contact's anniversary month
ad_d	Contact's anniversary day
bd_y	Contact's birth year
bd_m	Contact's birth month
bd_d	Contact's birth day

Adding LDAP Search Fields

You can add more fields to an LDAP search on the **Find People** page in the address book.

The `addrbook/locale/localename/cell.var` file for Address Book contains a variable, **ldapfilter**, whose value is the format of the pattern for an LDAP search. If this variable is branded and the search form on the HTML page, `search.html`, is augmented with new fields, Address Book will report the new field values into the pattern.

Enable this functionality with these two steps:

1. Add a new field to the search form by using the variable **moreldapsearchfields** in `cell.var`. This variable is similar to the **moreaddrsearchfields** variable.
2. Augment the pattern variable in the `cell.var` file, **ldapfilter**, with the new field, by using a variable reference `$(ldapfilter_varname)` where *varname* is the field name.

Example: To add “Company” (o) to the LDAP search functionality, the `cell.var` file must be modified like this:

```
ldapfilter="( (&(| (objectclass=person) (objectclass=inetorgperson)
(objectclass=organizationalperson)) (mail=$(mail)) (o=$(ldapfilter_o))
(| (cn=$(cn)) (sn=$(sn)) (givenname=$(givenname)))) )";

moreldapsearchfields="<tr>
<td align=right nowrap><font size=-1><b>Company:</b></font></td>
<td><font size=-1><input type=text name=\"o\" value=\"$(o)\" size=15
MAXLENGTH=\"30\"></font></td>
<td align=right nowrap><font size=-1><b>&nbsp;</b></font></td>
<td><font size=-1>&nbsp;</font></td>
<td align=right nowrap><font size=-1><b>&nbsp;</b></font></td>
<td><font size=-1>&nbsp;</font></td></tr>";
```

If you are not sure that your LDAP server will know the attribute you're adding, with the same value of **moreldapsearchfields**, you can define:

```
ldapfilter="( (&(| (objectclass=person) (objectclass=inetorgperson)
(objectclass=organizationalperson)) (mail=$(mail)) $(ldapfilter_o)
(| (cn=$(cn)) (sn=$(sn)) (givenname=$(givenname)))) )";

ldapfilter_o = "(o=$(ldapfilter_o))";
```

This way, a search on the attribute (o) is performed only if the user enters a value.

Please Note:

- ◆ The value entered by a user on any field of the search form is modified by adding an asterisk (*) at the beginning and end of the value, if it does not already contain this character, so that a simple value becomes a substring pattern. This is the current behavior. The value is also modified so that any OR character is escaped to ensure the LDAP search pattern is valid.
- ◆ By branding the LDAP search page, you can allow a user to perform a search on any fields, but only the fields supported by Address Book will display. Those fields are:
 - ❖ For each contact:
 - full name

nickname
email address
last name
first name
display name
work phone number
home phone number
cell phone number
pager phone number
fax phone number
organization/company
organizational unit/department
job title
company or home website
street address
location/city
state
zip code
country
note
year
month
day in month
year

- ❖ For each group:
 - group-name
 - contactN-cn
 - contactN-mail

See also “LDAP Search” in cell.var for more details.

Next Steps

Once you have branded the files, you must zip the files and publish them. How to do this is described in [Publishing a Brand](#) on page 103. Additionally, you can access the Administration Suite online help for more information. Go to **System > Branding > Help**.

Junk Mail Manager Branding

Junk Mail Manager (JMM) branding consists of changing any of the files that make up the JMM interface. The JMM interface structured similarly to the Standard Edition interface, and as such, much of the branding techniques are similar. The JMM application can be used in conjunction with Standard Edition WebMail and Corporate Edition WebMail.

Branding details about the Standard Edition version of the WebMail/WebCal GUI are in [Chapter 4, Standard Edition Branding](#).



Read [Chapter 1, About Branding](#) for important introductory information.

This chapter describes the files available for JMM branding and provides some application-specific branding tasks, including the following tasks:

- ◆ [Common JMM Branding](#)
- ◆ [Branding the Junk Mail Manager Pages](#)
- ◆ [Next Steps](#)



There are many variables, hidden input types, and code that you should not change. Doing so may break your application. This chapter describes the elements, and variable values that you can change and offers guidelines for HTML changes that you can make.

Common JMM Branding

This section describes the most common branding tasks available for Junk Mail Manager (JMM).

Detailed information on all the files that make up JMM, and additional available branding tasks, are provided in the following sections, beginning with [Branding the Junk Mail Manager Pages](#) on page 98.

Branding the Style Sheets

The JMM style sheet is located in the `extras/css/jmm_style.css` directory of the system brand zip file. Brand the JMM style sheet only when there is a need for some element to look different.

Find more information about style sheets in [Working with the Cascading Style Sheets](#) on page 20.

Branding the Login Page

You can customize the JMM login.html file with specified colors, a different logo, and links (Figure 26). This section describes changing the JMM Login page defined in the `spam/rg_login.html` file.



Typically, JMM runs on a RazorGate appliance, which uses the `rg_login.html` file. If your JMM is running on a Message Server, an additional Login page file, `spam/login.html`, is used.



Figure 26 Junk Mail manager Login Page Example

Junk Mail Manager Login Images

To change the logo, located in `spam/images` directory, replace the `logosmall.gif` image. This logo has width=142 height=118.



Do not remove or replace the `poweredby.gif`. The `poweredby.gif` logo should be left on the Login page. There are some options as to which `poweredby.gif` you use, see [Powered by Mirapoint](#) on page 26 for details.

To change the background colors, change the style.css selectors “genLogin” (text and background color) and “genLoginTitle” (text only) to the colors you want.

You can add a background image using the HTML `background` tag to the `body` class rule; for example:

```
<body class="genLogin"
onload="document.forms[0].elements[0].focus()"$(miravar_genLogin)
background="/images/login_background.gif">
```

Maintain the fallback so that your specified colors display even in a non-CSS enabled browser, open the `brandvars.var` file using a text editor and replicate your declarations as the appropriate values. For example, if you change the “genLogin” background in to light blue (`#ccffff`) also change the `$(miravar_genLogin) bgcolor` value in `brandvars.var` to `#ccffff`. The `brandvars.var` variables `$(miravar_genLogin)` and `$(miravar_genLoginTitle)` are fallbacks for “genLogin” and “genLoginTitle”.

Junk Mail Manager Login Text

The “User” and “Password” text are defined in `spam/rg_login.html`. You can change those words, but do not change the login form (also defined there).

The locale version footer is defined based on the locales that you have added to your system (through the CLI). You cannot change the names of the locales, but

you can use “genLogin” in the `extras/css/jmm_style.css` to change the colors, font, and background colors.

To add a link:

1. Create the introduction/information web page that you want to be available to people from the **Login** page and save it to the top-level `extras` directory.
2. Open a text editor and open the desired `login.html` or `rg_login.html` file.
3. Create a new link by adding an `href` tag referencing your non-system web page.

Branding the Banner

By default the JMM banner is blank, however, images and text can be added to the Junk Mail Manager banner using the `spam/banner.html` file.

The following branding variables (Table 35) and procedures apply to the JMM banner using the `spam/banner.html` file.

Table 35 banner.html Variables

	Variable	Defined in...	Comments
BODY class	genNav	CSS and brandvars.var	Text and background colors
Apps Menu	\$(appsmenu)	cell.var (key: appsmenu_tb)	<ul style="list-style-type: none"> ❖ WebMail and WebCal both licensed ❖ There is a placeholder for a logo
	genDataOff	CSS and brandvars.var	Disabled links text color
	genNavLink		Colors for regular, active, visited, and hover. Text-decoration : none.

- ◆ To change the background color, modify the “genNav” class in the `extras/css/jmm_style.css` file. Link styles, “genDataOff” and “genNavLink”, are also defined in the `extras/css/jmm_style.css` file.
- ◆ To add an image, place the image file you want to add in the `spam/images` directory. At the top of the `banner.html` file, beneath the HTML `body class` rule, add the `img src` tag as follows:


```

```
- ◆ To scale the image, specify **width** and **height** variables within the `img src` tag. For example:


```

```

Branding the Links Frame

The Links frame (Figure 27) uses the `spam/links.html` files; and displays on every page in Frames mode. The Links frame is always used in Junk Mail Manager.



Figure 27 Junk Mail Manager Links Frame Example

Links frame branding involves the use of the following variables (Table 36):

Table 36 `links.html` Variables

	Variable	Defined in...	Comments
BODY class	genNav	CSS and brandvars.var	Text and background colors
Primary Links	N/A	links.html	HREF links (except as noted below); uses <small> tag
	\$(compact_link) \$(emptytrash_link) \$(getmaillink)	cell.var (system key)	The system turns on or off these links depending on preference settings
	\$(helppath)	cell.var (system key)	Path to the online Help files
	genNavLink	CSS and brandvars.var	Colors for regular, active, visited, and hover. Text-decoration : none
Legend	N/A	links.html	“Legend” text font size=-1 Icon label text font size=-2
	\$(legend_priority)	cell.var (key: prioritycount_3or5)	How many Priority symbols display
	\$(deletelegend)	cell.var (key: deletelegend)	If Trash is turned off, this Legend displays with the Delete icon
	genHeader	CSS and brandvars.var	Header row color for table
	genLegend		Background color for table

Adding a Linked Advertisement

You can add a graphic image to your pages to advertise another Web site, providing direct link access. This procedure describes how to add a linked advertisement to the list of primary links defined in the `links.html` file in the `spam` directory for Junk Mail Manager. You can also apply this procedure to the `banner.html` file

To add an advertisement to that file instead of the `links.html` file:

1. Place the advertisement image file in the appropriate `images` directory.



Use `/wmi` as an alias to `/mail/images`.

2. At the top of the `links.html` file, beneath the HTML `body class` rule, use an `img src=` tag in conjunction with an `a href=` tag, as shown in the following example. Specify a URL that points to the web server where the rotating images reside:

```
<a href="http://www.webserver/image_name.html">
</a>
```

Substituting Button Images for Text Links

This procedure describes how to substitute button `.gif` images for the linked text defined in the `links.html` file in the `spam` directory for Junk Mail Manager.

1. Place the `.gif` files for the buttons into the appropriate `images` directory.
2. In the `links.html` file, locate the lines of text between the `<small>` and `</small>` tags. These lines provide links to their respective JMM windows and panes. The first line in this series specifies checking for new spam:

```
<small>
  <a class="genNavLink" href="/wm/mail/
  checkmail.html?sessionid=$(sessionid)&op=checkmail&mbox="
  target="rhs">Check Mail</a>
</small>
```

3. To add a button image for checking new spam, substitute the following for **Check Mail** in the line.

```

```

4. Repeat for each button image you want to use.

Branding the Help Files

The Junk Mail Manager Help files are here:

`help/locale/locale name/spam/`: Junk Mail Manager Help files.

Junk Mail Manager Help files use the `jmm_style.css` style sheet in the `extras/css` directory.



The URL that is shown at the bottom of the browser window when an online help page is displayed is: `/help/apps/locale`. Do not create a directory to mimic the hierarchy shown in the browser window. Keep the structure of the original zip file: `help/locale/locale name/`.

Branding the Junk Mail Manager Pages

This section provides an overview of the files in the `spam/locale/localename/` and `spamu/locale/localename/` directories that define the pages for Junk Mail Manager (Figure 28).



Junk Mail Manager does not have viewing mode options, all the pages display in **Frames** mode, in a frame along with the `spam` Banner (banner.html) and Links (links.html) frames.

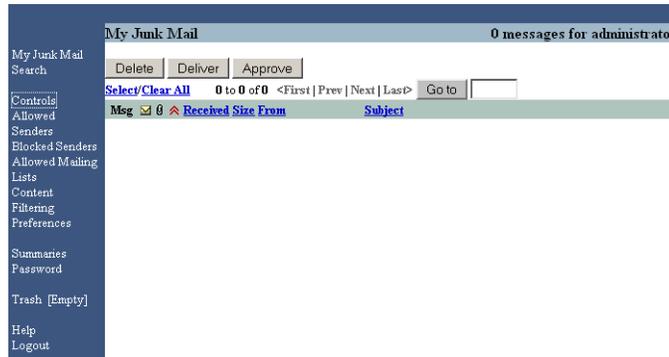


Figure 28 Junk Mail Manager Homepage Example

About the Junk Mail Manager Pages

The Junk Mail Manager pages are located in a separate directory called `spam/locale/localename/`. The pages are created by a combination of process files that manage requests, and HTML files that provide templates for the display. The HTML template files reference a CSS (cascading style sheet) located in the top-level `extras` directory. The style elements defined in the CSS are also defined in a special process file, `brandvars.var`, so browsers that do not support CSS still display specified style formatting.

Spam Frameset Definition Files

Table 37 describes the Junk Mail Manager frame definition files. These files are used in the **Frames** and **Frames with JavaScript** modes. You may need to adjust the frame set definitions in these files if you reorder elements or add graphics.

Table 37 Junk Mail Manager (spam/) Frameset Definition Files

Template File	Description & Dependencies
blank.html	A frame definition file that ensures that the correct files display in the correct order.
checkmail.html	References toolbar.html and inner.html for the My Junk Mail folder.
inner.html	References toc.html and read.html, or read_blank.html when no message is selected, for the My Junk Mail folder.
search.html	References searchform.html, toolbar.html, and inner.html for the Search page.

Table 37 Junk Mail Manager (spam/) Frameset Definition Files (Continued)

Template File	Description & Dependencies
window.html	Controls which files are displayed and how much room they take up. References banner.html, links.html and blank.html

Spam Process Files

These files are used to process system requests ([Table 38](#) and [Table 39](#)). You can make global changes by changing variable definitions in these files.

Table 38 Spam (spam/) Process Files

File Name	Description
cell.var	Contains variable definitions.
charsets.var	Contains character set name definitions.
locale.var	Contains customizable system and error messages.
timezones.var	Contains timezone name definitions.

Table 39 Spamu (spamu/) Process Files

File Name	Description
cell.var	Contains variable definitions.
locale.var	Contains customizable system and error messages.

Spam HTML Files

These are the template files ([Table 40](#) and [Table 41](#)). You can change element terminology, colors (in some cases), add graphics, and reorder data fields using these files.

Table 40 Junk Mail Manager (spam/) HTML Template Files

File Name	Description
checkmail_s.html	Creates the page that displays the toolbar buttons, the message list column headings, and message list for the My Junk Mail page.
confirmwindow.html	Confirmation page for preference settings requests.
logexpire.html	Creates the page that displays when a message link, in a Junk Mail Manager summary email, is clicked for a message that has expired and been deleted.

Table 40 Junk Mail Manager (spam/) HTML Template Files (Continued)

File Name	Description
login.html	Creates the Login page that opens when a user accesses the Junk Mail Manager URL, or clicks the Login link from within a Junk Mail Manager summary email on a Mirapoint Message Server. This page is fully brandable, you need only retain the system functions and login options. Note: Use rg_login.html if your JMM is on a RazorGate.
login_init.html	Creates the Junk Mail Manager cookies-authentication Login page.
logread.html	Creates the message contents page when you click the icon/subject from the JMM summary email
logread_blank.html	Creates a page for an empty message.
logreadrefresh.html	Recalls logread.html.
logsummary.html	Creates the log summary email that contains all the messages most recently categorized as spam.
logsummary_blank.html	Creates the log summary mail when there is no new spam since the last summary was sent.
nocookie.html	Cookies must be enabled page.
prefs_spam.html	Creates the Preferences page that opens when a user clicks Preferences .
read.html	Creates the Message List (TOC) frame.
read_blank.html	Displays empty contents for the message pane, when no message is selected.
read_s.html	Creates the page that displays your message contents.
refresh.html	Recalls read.html.
retry.html	Creates a “Unable to display your HTML attachment” page with directions.
rg_login.html	Creates the Login page that opens when a user accesses the Junk Mail Manager URL, or clicks the Login link from within a Junk Mail Manager summary email on a RazorGate. This page is fully brandable, you need only retain the system functions and login options.
search_s.html	Creates the page that displays the Searchform, the Toolbar, the Table of Contents column headings, and message list for the Search page.
searchform.html	Creates the Searchform frame.
timeout.html	Creates a “Session has timed out” page.
toc.html	Creates the page that displays the Table of Contents column headings for the My Junk Mail , and Search pages and the message list table.

Table 40 Junk Mail Manager (spam/) HTML Template Files (Continued)

File Name	Description
toolbar.html	Creates the Toolbar frame for the My Junk Mail , and Search pages.

Table 41 Junk Mail Manager (spamu/) HTML Template Files

File Name	Description
blacklist.html	Creates the Blocked Senders page.
changepass.html	Creates the Change Password page.
confirm_filter_delete.html	Creates the message filter Confirm Delete page.
confirmdelete.html	Creates the Confirm Delete page that opens when a user clicks Delete on the Allowed Senders , Blocked Senders , or Allowed Mailing Lists pages.
eu_contfilt_adv.html	Creates the Message Filters page.
eu_junkmail_filter.html	Creates the Controls page.
jmmlog_summary.html	Creates the Summaries page.
whitelist.html	Creates the Allowed Senders page.
whitelistto.html	Creates the Allowed Mailing Lists page.

Next Steps

Once you have branded the files, you must zip the files and publish them. How to do this is described in [Publishing a Brand](#) on page 103. Additionally, you can access the Administration Suite online help for more information. Go to **System > Branding > Help**.

Publishing a Brand

Publishing your brand allows you to view your brand and assign your brand to a domain.

This chapter provides information on the following topics:

- ◆ [Viewing Your Brand](#)
- ◆ [Zipping Your Brand Files](#)
- ◆ [Publishing Your Brand Files](#) on page 104
- ◆ [Assigning Your Brand](#) on page 104

Viewing Your Brand

Once you have made branding changes, you can view your brand using the `usage.html` file in `extras/css` (Standard Edition) or the `ce_usage.html` file in `extras/css` (Corporate Edition). Make sure the usage file is pointing to the right style sheet.

To test your brand, you must publish it.

Zipping Your Brand Files

In order to publish your modified brand files they must be compressed into one zip file.

To zip your brand files:

1. If appropriate, delete unmodified **help** and **locale** directories to make publishing and downloading easier. Once published, the default factory **help** and **locale** directory files automatically get used for those deleted directories.



Do not delete any other directories. MOS upgrading incompatibilities will emerge.

2. If you do not brand the Help files, delete the **help** directory.
3. Beginning your zip at the `Mirapoint_apps_x.x` level and maintaining the original directory structure, zip the modified files. See [Figure 1](#) on page 16 for a picture of the structure.



You cannot rename at this level, but you can rename at one level above; for example, `MyCompanyBrandx_x`.

Publishing Your Brand Files

Once your brand files have been modified and zipped, use the Administration Suite to publish your brand files.



You must enter a canonical name in your DNS server for each domain that you want assigned to a brand; this is a system administrator task and is described in the *Message Server Administrator's Guide*, and is also briefly described in the [Creating a Domain Sensitive Login Page](#) on page 27.

If you prefer to use the CLI for brand publishing, refer to *The Brand Command* chapter of the *Mirapoint Administration Protocol Reference Guide* for details.

To publish your brand:

1. Log in as an administrator and go to the **System > Branding > Publish** page.
2. Select the type of brand you are publishing, either **System** or **Named**.
Named brand—also enter a name for the brand.
3. Use the **Browse** button to locate your zipped custom brand file set and click **Publish**.

A status message displays the publishing results.

Once a brand is published, it displays on the **Branding > Download** page and may be selected on the **Branding > Select Brand** page.

4. For the publishing process to complete, stop and re-start WebMail and WebCal.
5. If there are errors, you must fix those and then publish again.

Once your brand is published, you can assign it to a domain.

Assigning Your Brand

Use the Administration Suite **System > Branding > Assign Brand** page to assign domains to your modified brands.

To assign a modified brand:

1. Using the **System > Branding > Select Brand** page, select your brand.
2. Go to the **Branding > Assign Brand** page and add domains to the selected brand. See the Administration Suite online Help for details.

You can now log into that domain and view your brand.

Troubleshooting

This section provides some problem solving information on topics such as error messages; URLs vs forms; how to view missing published changes; how to access the Administration Suite in cases of a “brand-out”; and, how to revert to the default system brand.

Error Messages

Following is a list of error messages, and suggestions:

- ◆ **This is not a zip file:** The file published is not a zip file. You must use a zip file.
- ◆ **Server busy, try again later:** The load on the server is too high, try again later.
- ◆ **The zip file’s uncompressed size is too large:** The file submitted is too big. The maximum uncompressed size is 50 MB; the maximum compressed size is 10 MB.

URLs Versus Forms

Some operations in WebMail or address book work only if they are using a certain executable format. For example:

- ◆ A URL link uses GET
- ◆ A form submission uses POST

While it is easy to convert a form to a URL, the operation might not work if the application expects a POST.

Duplicate Variable Keys

Always verify that there are no duplicate variable keys in the vartab files, because the last key of a given name will override any previously-found key(s).

Publishing Problem

If after publishing the brand, you do not see your changes (see [Chapter 6, Publishing a Brand](#)), it is possible that you forgot to stop and re-start WebMail and

WebCal to complete the publishing process. Stop and re-start WebMail and WebCal to see your newly published changes.

Sometimes a browser connection times out. If you suspect this may have happened, from the CLI, use the **brand add** command and FTP.

Failing to Display the Administration Suite Login

When logging into the Administration Suite is problematic, revert back to the default factory brand using the following commands:

Using the CLI, type the following commands:

```
telnet <host> 10144
brand delete system
```

Once the custom system brand is deleted, all domains use the default system brand.

Restoring the Factory Brand

Restoring the factory brand can be executed using either the Administration Suite or the CLI.

From the Administration Suite

To use the Administration Suite to restore the factory brand:

1. On the **Select Brand** page, under **System Brand**, click **Select**.
Your system brand is indicated in the lower left corner of the page and is available for modifications in the other **Branding** pages.
2. In the left side menu, click **Delete Brand**.
The **Delete Brand** page displays with a message at the top: “Restore to the factory default system brand.”
3. Click **Delete**.
Your custom system brand is reverted to the factory system brand. All domains assigned to **system** brand now have the factory default system brand restored to their interfaces.

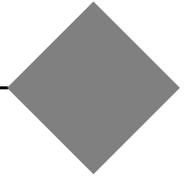
From the CLI

To use the CLI to restore the factory brand:

```
telnet <host> 10144
brand delete system
```

Once the custom system brand is deleted, all domains use the default system brand.

Index



A

- adding
 - a linked advertisement, Standard Edition & JMM [47, 97](#)
 - images [46, 95](#)
 - links to Corporate Edition banner [35](#)
- Address Book pages
 - about [84](#)
 - adding LDAP search fields [90](#)
 - branding [84](#)
 - branding options [87](#)
 - contacts variables [88](#)
 - localized import/export [87](#)
 - process files [85](#)
 - template files [85](#)
 - WebMail vs. WebCal [84](#)
- attachments, setting in WebMail [70](#)

B

- banner
 - Corporate Edition [35](#)
 - Standard Edition & JMM [45, 95](#)
- banner.html [54](#)
- branding
 - assigning domains [104](#)
 - configuring local hosts [15](#)
 - creating a domain sensitive login page [27](#)
 - creating delegated domains [14](#)
 - custom brand document [12](#)
 - downloading the factory zip file [15](#)
 - error messages [105](#)
 - getting ready [11](#)
 - Help files [41](#)
 - named brands, about [13](#)
 - process [12](#)
 - publishing your brand [104](#)
 - reverting back to the default [106](#)
 - setting up your workstation [13](#)
 - two types of variables [21](#)
 - using the usage.html file [20](#)

- variables format [23](#)
- variables, published, run, rendered [22](#)
- working with CSS [18](#)
- working with variables [21](#)
- branding files
 - cell.var, about [60](#)
 - HTML templates [66](#)
 - images [24](#)
 - locale.var, about [62](#)
- branding, Address Book
 - about [84](#)
 - localized import/export [87](#)
 - options [87](#)
- branding, Corporate Edition
 - banner [35](#)
 - login page [33](#)
 - navigation pane [36, 37](#)
 - options window [38, 39](#)
 - style sheets [31](#)
- branding, Corporate Edition branding [31](#)
- branding, JMM
 - style sheets [93](#)
- branding, Junk Mail Manager
 - process files [99](#)
 - template files [99](#)
- branding, Options/ Account pages
 - about [81](#)
 - process files [81](#)
 - template files [82](#)
- branding, Spelling pages
 - about [83](#)
 - process files [83](#)
 - template files [83](#)
- branding, Standard Edition
 - login page [44](#)
- branding, Standard Edition & JMM
 - adding a linked advertisement [47, 97](#)
 - adding images [46, 95](#)
 - banner [45, 95](#)
 - branding [43, 93](#)

- links.html [46](#), [96](#)
- login page [94](#)
- style sheets [44](#)
- substituting buttons for links [48](#), [97](#)
- branding, WebCal
 - about [71](#)
 - frameset files [74](#)
 - options [77](#)
 - process files [74](#)
 - template files [74](#)
- branding, WebMail
 - frameset definition files [59](#)
 - options [67](#)
 - process files [60](#)
 - template files [64](#)
- brands
 - assigning domains to [104](#)
 - named, about [13](#)
 - publishing [104](#)
 - two methods [13](#)
 - updating [25](#)
- brandvars.var
 - about CSS fallback [62](#)
 - CSS fallback plan [20](#)
- C**
- cell.var, about [60](#)
- charset
 - setting in WebMail [67](#)
 - WebMail, charsets.var [60](#)
- choose a resource [40](#)
- common resource names [23](#)
- configuring
 - domains [14](#)
 - workstation host file [15](#)
- Corporate Edition
 - banner [35](#)
 - branding [31](#)
 - branding the Help pages [41](#)
 - branding the themes [32](#)
 - login page [33](#)
 - login page images [33](#)
 - login page text [34](#)
 - navigation pane [36](#), [37](#)
 - options window [38](#), [39](#)
 - replacing the banner logo image [35](#)
 - style sheets [31](#)
 - upgrading to [26](#)
- creating
 - delegated domains for branding [14](#)
 - domain sensitive login pages [27](#)
 - variables [22](#)
- CSS
 - about usage.html [20](#)
 - fallback plan [20](#)
 - terminology [19](#)
 - working with [18](#)
- custom brand document, about [12](#)
- D**
- date format, setting in WebMail [68](#)
- default factory brand, reverting back to [106](#)
- defaults
 - mode settings, WebCal [77](#)
 - mode settings, WebMail [67](#)
 - reverting to the factory brand [106](#)
 - setting the date format, WebMail [68](#)
 - WebMail attachment handling [70](#)
 - WebMail charset definition [67](#)
- display_charset values [30](#)
- domain sensitive login page [27](#)
- domains
 - assigning to brands [104](#)
 - creating for branding [14](#)
- downloading
 - factory brand zip file [15](#)
 - Mirapoint powered-by logo [24](#)
- E**
- email messages, branding for WebCal [77](#)
- error messages, definitions [105](#)
- F**
- factory brand, reverting back to [106](#)
- fallback plan for the CSS [20](#)
- files, directory structure [16](#)
- Frames mode**
 - definition [50](#)
 - WebMail [50](#)
- Frames with JavaScript mode**
 - definition [51](#)
 - WebMail [51](#)
- frameset definition files
 - WebCal [74](#)
 - WebMail [59](#)
- H**
- Help pages, branding [41](#)
- I**
- images
 - branding [24](#)
 - replacing the banner, CE [35](#)

J

JavaScript files, **extras directory** 60

JMM

branding the style sheets 93

Junk Mail Manager

about 98

brand.vars, about CSS fallback 62

process files 99

template files 99

what can be branded 49

L

large group event scheduling limit 40

large group threshold 41

LDAP, adding search fields 90

links, adding to Corporate Edition banner 35

links.html 55

links.html, Standard Edition & JMM 46, 96

locale.var

about 62

definition 16

localized import/export, Address Book 87

login pages

branding, Corporate Edition 33

images, Corporate Edition 33

images, Standard Edition & JMM 45, 94

making domain sensitive 27

Mirapoint logo 24

text, Corporate Edition 34

text, Standard Edition & JMM 45, 94

M

Mirapoint

logo 24

modes

selecting 50

setting defaults, WebCal 77

setting defaults, WebMail 67

WebCal pages 71

WebMail template files 64

WebMail/WebCal 50

N

named brands, about 13

navigation pane, Corporate Edition 36, 37

No Frames mode

definition 52

WebMail 52

O

options window, Corporate Edition 38, 39

Options/ Account pages

about 81

branding 81

process files 81

template files 82

P

planning a custom brand 49

process files

Address Book pages 85

definition 16

Junk Mail Manager pages 99

Options/ Account pages 81

Spelling pages 83

WebCal pages 74

WebMail 60

publishing your brand 104

publish-time variables, definition 21

R

resource search enabled 40

reverting back to the default factory brand
106

run-time variables, definition 22

S

screen elements

custom branding 49

WebCal pages 73

WebMail 53

selecting a display mode 50

setting up branding environment 13

Spelling pages

branding 83

process files 83

template files 83

Standard Edition

branding 43, 93

contacts variables 88

files and directories 16

login page, quick branding 44

what can be branded 49

Standard Edition & JMM

adding a linked advertisement 47, 97

banner 45, 95

brand.vars, about CSS fallback 62

branding the Help files 41

branding the style sheets 44

links.html 46, 96

- login page images [45](#), [94](#)
- login page text [45](#), [94](#)
- login page, quick branding [94](#)
- substituting buttons for links [48](#), [97](#)
- style sheets
 - Corporate Edition [31](#)
 - JMM [93](#)
 - Standard Edition & JMM [44](#)
- substituting buttons for links, Standard Edition & JMM [48](#), [97](#)
- system variables, about [23](#)

T

- template files
 - Address Book pages [85](#)
 - Junk Mail Manager pages [99](#)
 - Options/ Account pages [82](#)
 - Spelling pages [83](#)
 - WebCal pages [74](#)
- template HTML files
 - branding [66](#)
 - definition [16](#)
- themes, branding [32](#)
- Three-Way Merge, branding upgrade technique [25](#)
- toc.html [58](#)
- toolbar.html [56](#)
- troubleshooting
 - about quotes [29](#)
 - Administration Login page not displaying [106](#)
 - duplicate keys [105](#)
 - error messages [105](#)
 - Help files URL [48](#), [97](#)
 - publishing problem [105](#)
 - URLs vs. Forms [105](#)

U

- upgrading
 - to Corporate Edition [26](#)
 - your brand [25](#)
- user tasks
 - logging in [11](#)
 - selecting a display mode [50](#)

V

- variables
 - about [21](#)
 - about quotes [29](#)
 - format [23](#)
 - published, run, rendered [22](#)

- system, about [23](#)
- two types [21](#)

- vartab files
 - definition [16](#)
 - per application [63](#)

W

- WebCal pages
 - about [71](#)
 - branding [71](#)
 - branding email messages [77](#)
 - branding options [77](#)
 - frameset definition files [74](#)
 - how they are defined [73](#)
 - modes [71](#)
 - process files [74](#)
 - screen elements [73](#)
 - setting default mode [77](#)
 - template files [74](#)
- WebMail
 - branding options [67](#)
 - frameset definition files [59](#)
 - how the pages are defined [59](#)
 - process files [60](#)
 - screen elements [53](#)
 - setting attachment handling [70](#)
 - setting default mode [67](#)
 - setting the charset [67](#)
 - setting the date format [68](#)
 - template files per mode [64](#)
 - upgrading to Corporate Edition [26](#)
- WebMail/WebCal modes [50](#)

Z

- zip
 - directory structure [16](#)
 - getting ready to publish [103](#)

MIRAPOINT SOFTWARE, INC. SOFTWARE LICENSE AGREEMENT

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT (“LICENSE”) CAREFULLY BEFORE DOWNLOADING OR OTHERWISE USING THE SOFTWARE. BY DOWNLOADING, INSTALLING OR USING THE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS SOFTWARE LICENSE AGREEMENT.

IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, YOU ARE NOT AUTHORIZED TO DOWNLOAD OR USE THIS SOFTWARE.

1. Scope. This License governs you (“User”) and your use of any and all computer software, any printed or electronic documentation, or other code, whether on disk, in read only memory, or on any other media (collectively, the “Mirapoint Software”) provided to you as part of or with a Mirapoint Product.
2. License, not Sale, of Mirapoint Software. The Mirapoint Software is licensed, not sold, to User by MIRAPOINT SOFTWARE, INC. or its affiliate, if any (“Mirapoint”). USER MAY OWN THE MEDIA ON WHICH THE MIRAPOINT SOFTWARE IS PROVIDED, BUT MIRAPOINT AND/OR MIRAPOINT’S LICENSOR(S) RETAIN TITLE TO THE MIRAPOINT SOFTWARE. The Mirapoint Software installed on the Mirapoint Product and any copies which this License authorizes the User to make are subject to this License.
3. Permitted Uses. This License allows User to use the pre-installed Mirapoint Software exclusively on the Mirapoint Product on which the Mirapoint Software has been installed. With respect to Mirapoint Software [identified by Mirapoint as the “administrative application”] that has not been preinstalled on the Mirapoint Product, this License allows you to copy, use and install such Mirapoint Software on one or more administrative workstations on which the Mirapoint Software is supported. User may make copies of the Mirapoint Software in machine-readable form for backup purposes only, provided that such backup copy must include all copyright and other proprietary information and notices contained on the original.
4. Proprietary Rights; Restrictions on Use. User acknowledges and agrees that the Mirapoint Software is copyrighted and contains materials that are protected by copyright, trademark, trade secret and other laws and international treaty provisions relating to proprietary rights. User may not remove, deface or obscure any of Mirapoint’s or its suppliers’ proprietary rights notices on or in the Mirapoint Software or on output generated by the Mirapoint Software. Except as permitted by applicable law and this License, you may not copy, decompile, reverse engineer, disassemble, modify, rent, lease, loan, distribute, assign, transfer, or create derivative works from the Mirapoint Software. Your rights under this License will terminate automatically without notice from Mirapoint if you fail to comply with any term(s) of this License. User acknowledges and agrees that any unauthorized use, transfer, sublicensing or disclosure of the Mirapoint Software may cause irreparable injury to Mirapoint, and under such circumstances, Mirapoint shall be entitled to equitable relief, without posting bond or other security, including but not limited to, preliminary and permanent injunctive relief.
5. Third Party Programs. Mirapoint integrates third party software programs with the Mirapoint Software which are subject to their own license terms. These license terms can be viewed at <http://www.mirapoint.com/licenses/thirdparty/eula.php>. If User does not agree to abide by the applicable license terms for the integrated third party software programs, then you may not install the Mirapoint Software.
6. Disclaimer of Warranty on Mirapoint Software. User expressly acknowledges and agrees that use of the Mirapoint Software is at your sole risk. Unless Mirapoint otherwise provides an express warranty with respect to the Mirapoint Software, the Mirapoint Software is provided “AS IS” and without warranty of any kind and Mirapoint and Mirapoint’s licensor(s) (for the purposes of provisions 5 and 6, Mirapoint and Mirapoint’s licensor(s) shall be collectively referred to as “Mirapoint”) EXPRESSLY DISCLAIM ALL WARRANTIES AND/OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN ADDITION, MIRAPOINT DOES NOT WARRANT THAT THE MIRAPOINT SOFTWARE WILL MEET YOUR REQUIREMENTS, OR THAT THE MIRAPOINT SOFTWARE WILL RUN UNINTERRUPTED OR BE ERROR-FREE, OR THAT DEFECTS IN THE MIRAPOINT SOFTWARE WILL BE CORRECTED. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES OR OTHER DISCLAIMERS, SO THE ABOVE EXCLUSION OR DISCLAIMERS MAY NOT APPLY TO YOU.
7. Limitation of Liability. UNDER NO CIRCUMSTANCES, INCLUDING NEGLIGENCE, SHALL MIRAPOINT BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO THIS LICENSE. FURTHER, IN NO EVENT SHALL MIRAPOINT’S LICENSORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES (INCLUDING BUT NOT LIMITED TO PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOSS OF USE, DATA OR PROFITS OR INTERRUPTION), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY (INCLUDING NEGLIGENCE OR OTHER TORT), ARISING IN ANY WAY OUT OF YOUR USE OF THE SOFTWARE OR THIS AGREEMENT, EVEN IF ADVISED OF THE POSSIBILITY OF DAMAGES. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OF



INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THIS LIMITATION MAY NOT APPLY TO YOU. In no event shall Mirapoint's total liability to you for all damages exceed the amount paid for this License to the Mirapoint Software.

8. Export Control. As required by the laws of the United States and other countries, User represents and warrants that it: (a) understands that the Mirapoint Software and its components may be subject to export controls under the U.S. Commerce Department's Export Administration Regulations ("EAR"); (b) is not located in a prohibited destination country under the EAR or U.S. sanctions regulations (currently Cuba, Iran, Iraq, North Korea, Sudan and Syria, subject to change as posted by the United States government); (c) will not export, re-export, or transfer the Mirapoint Software to any prohibited destination or persons or entities on the U.S. Bureau of Industry and Security Denied Parties List or Entity List, or the U.S. Office of Foreign Assets Control list of Specially Designated Nationals and Blocked Persons, or any similar lists maintained by other countries, without the necessary export license(s) or authorization(s); (d) will not use or transfer the Mirapoint Software for use in connection with any nuclear, chemical or biological weapons, missile technology, or military end-uses where prohibited by an applicable arms embargo, unless authorized by the relevant government agency by regulation or specific license; (e) understands and agrees that if it is in the United States and exports or transfers the Mirapoint Software to eligible users, it will, to the extent required by EAR Section 740.17(e), submit semi-annual reports to the Commerce Department's Bureau of Industry and Security, which include the name and address (including country) of each transferee; and (f) understands that countries including the United States may restrict the import, use, or export of encryption products (which may include the Mirapoint Software and the components) and agrees that it shall be solely responsible for compliance with any such import, use, or export restrictions.

9. Miscellaneous. This License will be governed by and construed in accordance with the laws of the State of California, U.S.A., without reference to its conflict of law principles. If a court of competent jurisdiction finds any provision of this License invalid or unenforceable, that provision will be amended to achieve as nearly as possible the same economic effect as the original provision and the remainder of this License will remain in full force. Failure of a party to enforce any provision of this License shall not waive such provision or of the right to enforce such provision. This License sets forth the entire agreement between the parties with respect to your use of the Mirapoint Software and supersedes all prior or contemporaneous representations or understandings regarding such subject matter. No modification or amendment of this License will be binding unless in writing and signed by an authorized representative of Mirapoint. You will not export, re-export, divert, transfer or disclose, directly or indirectly, the Mirapoint Software, Mirapoint Products or any technical information and materials supplied under this Agreement without complying strictly with the export control laws and all legal requirements in the relevant jurisdiction, including without limitation, obtaining the prior approval of the U.S. Department of Commerce.