MIRAPOINT®

Branding Guide

Release 4.0
September, 2007
Part Number: 010-00544d

This manual supports Messaging Operating System (MOS) release 4.0 and later MOS releases until replaced by a newer edition.

This manual and Mirapoint software are copyright © 1998-2007 Mirapoint, 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, 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, Inc. Mirapoint Message Server, Mirapoint Directory Server, Mirapoint Operations Console, RazorSafe, DirectPath, WebMail Direct, WebCal Direct, and GroupCal Direct are trademarks of Mirapoint, Inc.

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 © 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.

This software is derived in part from Red Hat Enterprise Linux, which is Copyright © 2005 Red Hat, Inc. All rights reserved.

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

The OpenLDAP Public License Version 2.8, 17 August 2003

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source form must retain copyright statements and notices,

2. Redistributions in binary form must reproduce applicable copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and

3. Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time. Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE 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 BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

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.

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.

MIRAPOINT, 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 LICENSE. 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 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 you by MIRAPOINT, INC. or its affiliate, if any ("Mirapoint"). YOU 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 you to make are subject to this License.

3. Permitted Uses. This License allows you 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 pre-installed 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. You may make one copy 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. You acknowledge and agree that the Mirapoint Software is copyrighted and contains materials that is protected by copyright, trademark, trade secret and other laws and international treaty provisions relating to proprietary rights. You 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. You acknowledge and agree 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. Disclaimer of Warranty on Mirapoint Software. You expressly acknowledge and agree 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.

6. 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.

7. Government End Users. If the Mirapoint Software is supplied to the United States Government, the Mirapoint Software and any documentation are provided with RESTRICTED RIGHTS. The Mirapoint Software is classified as "commercial computer software" and the documentation is classified as "commercial computer software documentation" or "commercial items," pursuant to DFAR Section 227.7202 or FAR Section 12.212, as applicable. Any use, modification, reproduction, display or disclosure of the Mirapoint Software or any documentation by the United States Government shall be governed by the terms of this License.

8. 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, reexport, 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.

Contents

List of Tables 9

List of Figures 11

Preface 13

Assumptions 13
About the Mirapoint Documentation 13
Typographic Conventions 14
Icon Conventions 14
Additional Branding Information 15
Getting Customer Support 15

1
Branding Basics, Tips, and Common Tasks 17

About Branding 17
Custom Branding vs. Quick Branding 17
Branding Files and Directories 18
System Brand vs. Named Brand 20
Branding Pitfalls 20
Branding System Requirements 21
The Branding Process 21
Planning Your Brand/Creating Your Custom Brand Document 22
Setting Up Your Branding Environment 22
Viewing, Zipping, Publishing, and Assigning Your Brand 24
Working with the Cascading Style Sheets 25
A Note About CSS Terminology 26
About usage.html and ce_usage.html 27
About the CSS Fallback: brandvars.var 27
Working with Variables 28
About Variables 28
Common Resource Names 30

Working with Image Files 31
Powered by Mirapoint 31
Tips 31
About Quotes 31
URLs versus Forms 32
Publishing Problem 32
Locale display_charset Values 32
Duplicate Keys 33
Troubleshooting 33
Common Tasks 34
Updating Brands 34
Upgrading from Standard Edition to Corporate Edition 35
Creating a Domain Sensitive Login Page 36
Branding the Login Page Using JavaScript 37

2
Quick Branding 39

Corporate Edition Quick Branding 39
Branding the Style Sheets—CE 39
Branding the Login Page—CE 41
Branding the Banner—CE 43
Branding the Navigation Pane—CE 44
Branding the Options Links—CE 44
Branding the Help Files—CE 46
Standard Edition/JMM Quick Branding 46
Branding the Style Sheets—SE and JMM 46
Branding the Login Page—SE and JMM 46
Branding the Banner—SE and JMM 48
Branding the Links Frame—SE and JMM 49
Branding the Help Files—SE and JMM 51
Quick Branding Next Steps 51

3
Custom Branding 53

Branding the WebMail Direct Pages 54
WebMail Addressbook 54
Planning Your Custom Brand 54
WebMail Viewing Modes 55
About the WebMail Screen Elements 59
Frame Files Reference 60
How the WebMail Pages are Defined 66
More WebMail Branding 73

Setting a Default Charset for WebMail 73
Branding Default Mode Preferences for WebMail 73
Changing the Date Format in WebMail 74
Changing How WebMail Handles Attachments 75
Branding the WebCal Direct Pages 76
About the WebCal Pages 76
WebCal Addressbook 77
About the WebCal Viewing Modes 77
About the WebCal Screen Elements 79
How the WebCal Pages are Defined 79
More WebCal Branding 83
Supporting Group Calendar 83
Branding Default Mode Preferences for WebCal 83
Branding Calendar Email Messages 83
Branding the Options/Account Pages 87
About the Options/Account Pages 87
Options/Account Process Files 88
Options/Account HTML Template Files 88
Branding the Spelling Pages 89
About the Spelling Pages 89
Spelling Process Files 89
Spelling HTML Files 90
Branding the Address Book Pages 90
About the Address Book Pages 90
Address Book—WebMail vs. WebCal 90
Address Book Process Files 91
Address Book HTML Template Files 91
More Address Book Branding 93
Localized Import/Export in Address Book 93
Branding Contact Data 94
Adding LDAP Search Fields 96
Branding the Junk Mail Manager Pages 97
About the Junk Mail Manager Pages 97
Spam Frameset Definition Files 98
Spam Process Files 98
Spam HTML Files 99
Custom Branding Next Steps 100
Index 101

List of Tables

Table 1 Typefaces Used in This Developer's Guide....14
Table 2 Icons Used in This Book....14
Table 3 Example Multiple Brand Environment....23
Table 4 Example of the CSS Fallback for the genHeader Class....27
Table 5 banner.html Variables....48
Table 6 links.html Variables....50
Table 7 Frames with JavaScript Mode Example....56
Table 8 Frames Mode Example....58
Table 9 No Frames Mode Example....59
Table 10 banner.html Variables....60
Table 11 links.html Variables....62
Table 12 toolbar.html Variables....63
Table 13 toc.html Variables....65
Table 14 WebMail (mail/) Frameset Definition Files....66
Table 15 WebMail (mail/) Process Files....66
Table 16 Other vartab Files used by the Applications....69
Table 17 WebMail (mail) HTML Template Files....70
Table 18 Default Preference Keys Explained....73
Table 19 ANSI C Function strftime Format Directives....75
Table 20 WebCal Frameset Definition Files....80
Table 21 WebCal Process (mcal/) Files....80
Table 22 WebCal (mcal/) HTML Template Files....80
Table 23 WebCal Default Preference Keys Explained....83
Table 24 Email Messages Defined in mcaldmsg.var....84
Table 25 Options/Account (user/) Process Files....88
Table 26 Options/Account (user/) HTML Template Files....88
Table 27 Spell (spell/) Process Files....89
Table 28 Spelling (spell/) HTML Template Files....90
Table 29 Address Book (addrbook/) Process Files....91
Table 30 Address Book (addrbook/) HTML Template Files....91
Table 31 Contact Table Row Format Variables....94
Table 32 Contact Table Row Data Variables....94
Table 33 Junk Mail Manager (spam/) Frameset Definition Files....98
Table 34 Spam (spam/) Process Files....98
Table 35 Spamu (spamu/) Process Files....98
Table 36 Junk Mail Manager (spam/) HTML Template Files....99
Table 37 Junk Mail Manager (spamu/) HTML Template Files....100

List of Figures

Figure 1 System Brand Zip Directory Structure (must be maintained) 18
Figure 2 Powered by Mirapoint Logo 31
Figure 3 Corporate Edition Login Page Example (text variables explained next) 41
Figure 4 Corporate Edition Banner Example 43
Figure 5 Corporate Edition Navigation Pane Example 44
Figure 6 Corporate Edition Options Window 45
Figure 7 Standard Edition WebMail Direct Login Page Example 47
Figure 8 Standard Edition Banner Example 48
Figure 9 WebMail Standard Edition Links Frame Example 49
Figure 10 Frames With JavaScript Mode, WebMail Standard Edition 56
Figure 11 Frames Mode, WebMail Standard Edition 57
Figure 12 No Frames Mode, WebMail Standard Edition 59
Figure 13 Banner, Calendar highlighted 60
Figure 14 Links Frame, WebMail on the left, WebCal on the right 61
Figure 15 toolbar.html Frames with JavaScript Mode, Delete to Trash *disabled* *63*
Figure 16 toolbar.html Frames with JavaScript Mode, Delete to Trash *enabled* *63*
Figure 17 toc.html Frames with JavaScript Mode 65
Figure 18 WebCal Screen Elements—Frames Mode 78
Figure 19 WebCal Screen Elements—No Frames Mode 79
Figure 20 Address Book Simplelinks Menu for WebMail Direct 91
Figure 21 Address Book Simplelinks Menu for WebCal Direct 91

Preface

Welcome to the Mirapoint *Branding Guide* for release 4.0. This book is intended for administrators of a Mirapoint messaging system and provides the following information:

- Chapter 1, Branding Basics, Tips, and Common Tasks—Branding basics and background information you need including important information on using the style sheets, the variables, the image files, upgrading to Corporate Edition, plus branding tips, troubleshooting, and common tasks.
- Chapter 2, Quick Branding—WebMail/WebCal Corporate Edition, WebMail/WebCal Standard Edition, and Junk Mail Manager branding tasks: changing the style sheets, the **Login** page, banner, navigation or links frame, and online Help files.
- Chapter 3, Custom Branding—A file reference, with some specific branding tasks, for WebMail/WebCal Standard Edition, including the Standard Edition **Options** pages (which are also the Administration Suite **Account** pages for end-users), the Standard Edition **Spelling** pages, the Standard Edition **Address Book** pages, and Junk Mail Manager.

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 the Mirapoint Documentation

Documentation for all Mirapoint products is available through the Mirapoint Technical Library (MTL) on the Customer Support website:

http://support.mirapoint.com/secure/MTL/MTL

The MTL 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 you have a valid Support Contract but need a Support login ID, send an email to:

support-admin@mirapoint.com

Typographic Conventions

Table 1 explains what different fonts in this book indicate.

Table 1 Typefaces Used in This Developer's Guide

Typeface	Use	Example
Verdana, Arial, Helvetica, sans-serif	Ordinary text	Which antispam scanner you use depends on which licenses you have applied.
Verdana, Arial, Helvetica, sans-serif Bold	Screen elements such as page titles, option labels, and commands	Follow these steps on the **Add User** page.
Verdana, Arial, Helvetica, sans-serif Italic	Placeholders for variables you provide	*your_IP_address*
`Typewriter`	Screen display text and command names	`Enter your IP address`
`Typewriter Bold`	Text that you type exactly as shown	**`User List`**
`Typewriter Italic`	Placeholders for variables you provide in text that you type	*`your_IP_address`*

Icon Conventions

Table 2 explains what the different icons in this book indicate.

Table 2 Icons Used in This Book

Icon	Indicates
	Important information that you should note.
	Critical information.
	Best practices.

Additional Branding Information

Additional branding information and product information can be found online. To access this information, log in to the Administration Suite as an administrator and go to the Help section called **Branding**. For general WebMail information, access the WebMail Help.

Getting Customer Support

If you experience problems with your system, contact Customer Support by email or by telephone:

Email: support@mirapoint.com

Telephone: 1-877-MIRAPOINT (647-2764)

When contacting Customer Support, please be prepared with the following information about your system:

- MOS version (**Version** command in the CLI)
- Host ID (**License Hostid** command)
- Serial Number (**Model Get Serial** command)
- Hardware model (**Model Get Chassis** command, but double-check for the hardware model on the front panel of the system)

1 Branding Basics, Tips, and Common Tasks

Branding allows you to change the look of the application web pages to match your company look. With custom branding you can also add or remove functionality, images, URL links, and so forth. Branding is available for WebMail/WebCal Corporate Edition and Standard Edition, as well as Junk Mail Manager.

This chapter includes these topics:

- About Branding
- The Branding Process
- Working with the Cascading Style Sheets
- Working with Variables
- Working with Image Files
- Tips
- Common Tasks

This book is intended to supply enough information so that the branding professional can make any acceptable changes to the application web pages that he or she wants. You have more branding options with Standard Edition and Junk Mail Manager than you do with Corporate Edition.

About Branding

This section provides basic information on Mirapoint branding.

Custom Branding vs. Quick Branding

WebMail/WebCal Standard Edition, as well as Junk Mail Manager, are custom brandable, which means you have access to the HTML template files as well as the cascading style sheet (CSS), vartab, and process files.

WebMail/WebCal Corporate Edition is quick brandable; you have access only to the CSS (and special "theme" CSS files); one vartab file, brandvars.var; login.html and banner.html in the **eml** directory and .gifs in **images/** and **images/eui**.

Branding Files and Directories

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

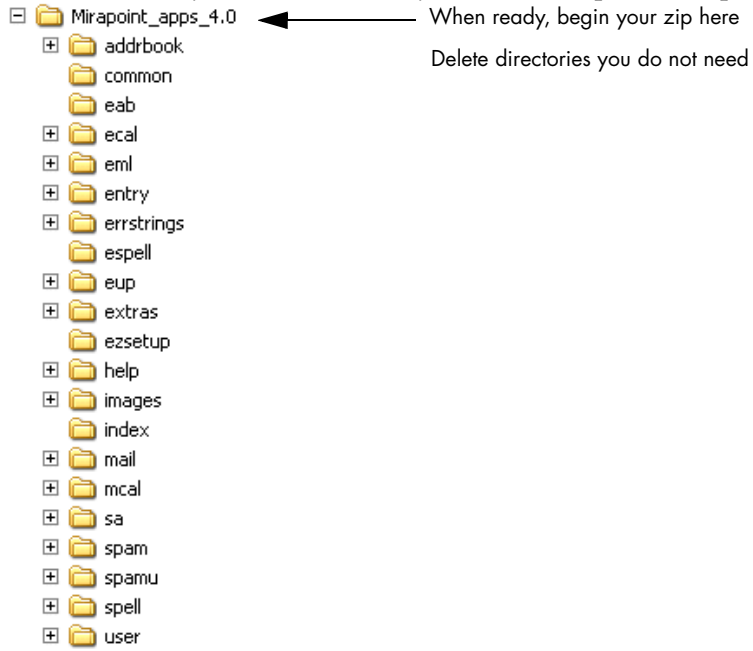


Figure 1 System Brand Zip Directory Structure (must be maintained)

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 **vartab** (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 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.

These style sheets: admin.css, rg_style.css, rgez_style.css, sa45_admin.css, styleMpHelp.css, styleMpHelpHome.css, styleRgHelp.css, and style.RgHelpHome.css are used for the Administration Suite pages for which branding is not supported.

- 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 Quick Branding WebMail/WebCal Corporate Edition.
- 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, these 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/

Some of these directories contain a **locale** directory; you can delete any locales contained therein that you do not need in your brand.

If you are not branding the Help files, delete the **help** directory; online Help will still be available.

System Brand vs. Named Brand

There are two ways to publish your customized factory system brand; they are:

- **System brand**: All domains not assigned to a named brand automatically use the custom "system" brand that you publish. Publish your own custom system brand by selecting **System Brand** on the **System > Branding: Publish** page of the Administration Suite after you have customized the application file set and are ready to upload it. If you do not publish a custom system brand, all domains not assigned a named brand use the factory system brand.
- **Named brand**: Named brands provide the ability to have different **Login** pages per domain; the over quota message can be set per brand or per domain; named brands can be localized. Named brands use a file set identical to the system brand. Publish any number of named brands by selecting **Named Brand** on the **System > Branding: Publish** page. Give the brand a name, and assign any number of domains to a named brand.

Branding Pitfalls

It is very important to *not* do any of the following:

- Attempt to publish only a few files. While entire directories can be deleted, the original file structure must be maintained in the zip; see Branding Files and Directories on page 18 for an illustration.
- Change the name of a file, your brand will break or the factory value will be used.
- Create any symbolic links, they are deleted on publishing.
- Attempt to publish zip files larger than 30 MB, they are rejected.
- Put custom JavaScript files in an application directory (for example, **mail**). The system looks for custom JavaScript files in the **extras** directory; if they are in a different directory they will not be served.
- Mix together components from different brands and/or different MOS releases; this can cause issues. For example, using a brandvars.var or css file from a previous MOS or other system brand file set will not work.

Branding System Requirements

Before you start branding, you need the following:

- A Mirapoint System that is installed with the appropriate licenses for the applications that you want to brand
- The hostname and HTTP access to the Mirapoint system
- System administrator login and password

To see the user application pages, enter

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, enter

http://*hostname*/acctadmin

The Branding Process

To publish a brand, you must download a zip file containing the factory application file sets of HTML templates, vartab and other process files, image and online help files; and then customize the files. The branding process includes several procedures.

First: Plan your brand and begin creating a Custom Brand document for each brand (system and named) that you want to create. Include all color, image, text, and other changes that you want to make. For Standard Edition branding, determine if you want to brand all three modes; often the **No Frames** mode is left un-branded. A Custom Brand document is very helpful when upgrading. Information on doing this is described in this section; see Planning Your Brand/Creating Your Custom Brand Document on page 22.

Second: Setup your branding environment. Create delegated domains, and configure your local host file so you can access them. This is so you can manage the creation of multiple brands. How to do this is described in this section. Make a secure copy (preferably on another system) of your system application file set. Download the factory brand zip file, unzip the file, and make a copy of the file set for each brand you want to make. Information on doing this is described in this section; see Setting Up Your Branding Environment on page 22.

Third: Customize the files. This is the most extensive task. See Chapter 2, Quick Branding and ” and/or Chapter 3, Custom Branding” for detailed for information. Test your new brands out to make sure all elements of the brand function properly and render properly.

Fourth: Publish your custom system brand and named brands, and assign domains to named brands, using the Administration Suite **System > Branding** pages.

Information on doing this is described in this section; see Viewing, Zipping, Publishing, and Assigning Your Brand on page 24.

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 Common Tasks section at the end of this chapter.

Planning Your Brand/Creating Your Custom Brand Document

As you plan your brand, create your Custom Brand document: your record of the changes you make to create a 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. Use the reference chapters of this book (when custom branding) to 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.

Setting Up Your Branding Environment

The following instructions are for setting up your workstation for branding. You need the following:

- A Windows or UNIX workstation
- A zip utility installed on the workstation
- An HTML editor. A simple text editor is the safest. You can use your choice of HTML editors; however, they must not add any extraneous information or change the format of the files. Do not use MS Wordpad™ because it can cause file format changes that result in errors when the files are published. Notepad™ and Dreamweaver™ are two HTML editors that work well.

Follow either the one brand or multiple brand setups, described next, depending on how many brands you plan to do at one time.

One Brand Setup

If you plan to work on one brand at a time, there is no additional setup. One Brand Setup uses brands published and viewed as System brands. System brands are discussed in System Brand vs. Named Brand on page 20. If this is your plan, you can skip to Downloading the Factory Brand on page 24.

Multiple Brand Setup

If you plan to work on more than one brand at a time, or if more than one person is working on brands at the same time, follow the multiple brand setup. This involves creating delegated domains on the Mirapoint system and either configuring the

hosts file on your workstation to access the domains, or configuring a DNS server as you would in a production environment.

The Multiple Brand Setup uses brands published and viewed as Named brands, which allows you to create brands, and assign domains to them. Named brands are viewed by logging into different domains using the at sign (@) and domain name in the **Username** login box.

Begin by creating some delegated domains; then, configure the local hosts. Both procedures are described in this section, next. Table 3 describes a possible configuration scheme that you could use to set up multiple delegated domains; the procedure to do so follows.

Table 3 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, follow these steps. An example configuration is shown in Table 3.

1. As an administrator logged in to the Administration Suite, click **Domains** and then **Administration**.
   Result: The **Domain: Administer Domains** page displays.
2. Enter the following data, see Table 3 for examples:
   - **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).
   - **Domain Disk Quota** *(Optional)*: The amount of available domain-wide disk space (in KB) for the delegated domain.
   - **Maximum Users** *(Optional)*: 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.

   Click **Add Domain**.
   Result: 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. You will then be able to view multiple named brands on one server without setting up a DNS server. The

hosts file must be configured on every workstation where the domain brand will be viewed.

On a Windows system, open C:\WINNT\system32\drivers\etc\hosts. On a UNIX system, open the comparable file; for example, '/etc/hosts'.

Enter the IP address of the Mirapoint system, and the domains you created in the format specified by the file, then save the file.

Downloading the Factory Brand

Use the Administration Suite to download the factory brand zip file. To do this, log in as an administrator and go to **System > Branding > Download**. Select the brand file appropriate for your system, DOS or UNIX, and click **Download Selected File**. Save and un-zip the file. In the CLI, use **Brand Get *brandname*** to retrieve previously added brands.

At this point, you customize the files. How to do this is described in detail in the following two chapters, Chapter 2, Quick Branding, and Chapter 3, Custom Branding.

Viewing, Zipping, Publishing, and Assigning 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); just make sure the usage file is pointing to the right style sheet. To test your brand, you must first publish it.

In order to publish your brand, you must zip the modified files, being careful to maintain the original directory structure; see Figure 1 on page 18 for a picture of the structure. Begin your zip at the **Mirapoint_apps_*x.x*** level; you cannot rename at this level, but you can rename at one level above; for example, ***MyCompany*Brand*x_x***. You can then publish the brand, described next; and assign a domain to it, also described in this section.

Do not use WinZip™ to zip the files.

Use the Administration Suite to publish your brand files, once they have been modified and zipped. To do this, login as an administrator and go to the **System > Branding > Publish** page. Select the type of brand you are publishing, either **System** or **Named**. If publishing a named brand, also enter a name for the brand. Use the **Browse** button to locate your zipped custom brand file set and click **Publish**; a message displays indicating the results. Once a brand is published, it displays on the **Branding > Download** page and may be selected on the **Branding > Select Brand** page.

In the CLI, use **Brand Add *brandname*** to publish new brands; **Brand Change *brandname*** to re-publish previously added brands.

Sometimes stopping and re-starting WebMail and WebCal is needed for the publishing process to complete.

To make publishing and downloading easier, delete all extraneous directories before zipping up your custom brand. Refer to page 19 for a list of the minimum required directories per application. If you do not brand the Help files, delete the **help** directory. Later, you can download your custom brand and modify only the files you need, zip it back up—with **Mirapoint_apps_*x.x*** as the starting point—and re-publish it.

If there are errors, you must fix those and then try publishing again. Once your custom brand is published, you can assign it to a domain.

Use the Administration Suite **System > Branding > Assign Brand** page to assign domains to brands. To do this, first select the brand using the **System > Branding > Select Brand** page, then go to the **Branding > Assign Brand** page and add domains to the selected brand. See the Administration Suite online Help for details. In the CLI, use **Brand Assign** ***domain brandname***.

You can now log in to that domain and view your brand.

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).

A Note 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, it is helpful to use usage.html (Standard Edition) and ce_usage.html (Corporate Edition) to view the CSS branding changes you would like to make. The usage.html file references style.css; ce_usage.html references uui.css and themes/system-default.css. To access usage.html and/or ce_usage.html, use these URLs:

- For Standard Edition:
  `extras/css/usage.html`
- For Corporate Edition:
  `extras/css/ce_usage.html`

About the CSS Fallback: brandvars.var

Some older web browsers do not support CSS, or they allow CSS support to be disabled. To ensure that your branded web pages display correctly without CSS, a special vartab file, brandvars.var is available. For each class defined in the style sheets there is a corresponding miravar_ variable defined in the brandvars.var file. This feature is known as the “CSS Fallback,” it ensures that the style sheet definitions are implemented in no-style-sheet browsers. This same file is also used by Corporate Edition Quick Branding, for some changeable variables. The brandvars.var file works as follows for the Custom Branding fallback:

- 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, allowing you to define your own variables.
- The substitutions declared are performed at brand publish-time to eliminate human error.

Table 4 Example of the CSS Fallback for the genHeader Class

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

Table 4 Example of the CSS Fallback for the genHeader Class

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

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)">...
```

Always use the brandvars.var file associated with the MOS version that you are branding. Do not re-use a previously branded one.

There is a bug in Netscape 4.X that is exposed if the above-mentioned recommendation is not followed. The Netscape 4.X bug is that 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 this bug will not be exposed.

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 Quick Branding is brandvars.var.

About Variables

Many text elements on the web pages are instantiated with the use of 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 Quick 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 $(dflttime) (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 (\).

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:

<img src="images/*imagename*.gif">.

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.

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

An example is shown in Figure 2, below.

Figure 2 Powered by Mirapoint Logo

Tips

This section offers some experience-won advice.

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 (\). These rules determine whether double quotes are required or not:

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

URLs versus Forms

Some operations in WebMail or address book work only if they are using a certain method. Usually:

- A URL link is a GET
- A form submission is a POST

Sometimes it is easy to convert a form to a URL but the operation will not work, because the application expects a POST.

Publishing Problem

Sometimes stopping and re-starting WebMail and WebCal is needed for the publishing process to complete.

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.

Duplicate Keys

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

Troubleshooting

This section provides some information on error messages as well as how to access the Administration Suite in cases of 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.
- **Uncompressed file size too large**: The file submitted is too big. The maximum uncompressed size is 50 MB; the maximum compressed size is 10 MB.
- **ERROR publishing submitted file**: This is a general error, check the screen for a specific error message.
- **Brand the username and/or the password are incorrect**: This is the response from the FTP server where the zip file is located. It is not set up properly for passive, or anonymous FTP.
- **Brand cannot re-brand**: A command was sent to WebMail and mcal, to flush cell.var data, and it failed. Either they were licensed, but not running, or they were busy, and did not respond. WebMail needs to be restarted. The user

should immediately see new templates, but information in the cell.var files will not be seen. Stopping and restarting WebMail and mcal will reflect the new date.

The Administration Suite Login page is not displaying, what can I do?

If you ever find yourself in a situation where you can't log in to the Administration Suite, here is how to revert back to the default factory brand.

Using the CLI:

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

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

How do I restore the factory brand?

If you are in the Administration Suite and want to restore the factory brand as default, follow these steps.

1. On the **Select Brand** page, under **System Brand**, click **Select**.
   Result: Your system brand is indicated in the lower left corner of the page and is available for modifications in the other **Branding** pages.
2. Click **Delete Brand** in the left side menu.
   Result: the **Delete Brand** page displays with a message at the top: "Restore to the factory default system brand."
3. Click **Delete**.
   Result: 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.

If you are using the CLI, restore the factory brand using this command:

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

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

Common Tasks

This section describes some branding tasks common to both Corporate Edition and Standard Edition/Junk Mail Manager.

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.

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 3.8, download the 3.8 factory brand and diff it against the 3.7 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 3.8 is a 3.6 brand, diff it against a 3.6 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.

If possible, create your brand primarily using the CSS. In this manner, you can change all of the colors used by the pages and add your own images (logos, and so forth) and future brand upgrades will be simple. A Custom Brand document is invaluable when upgrading.

Brands are not impacted by upgrading between dot-dots in a release, but Mirapoint can not guarantee compatibility when migrating between major releases.

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 3.7 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 Select *brandname* 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 structure (but with un-branded directories deleted)—and publish the brand using the **System > Branding > Publish** page as either your "system" brand or a "named" brand, see System Brand vs. Named Brand on page 20 and Viewing, Zipping, Publishing, and Assigning 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 36 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 these 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)

2 Quick Branding

Quick branding consists of changing the style sheets and brandvars.var variables, the **Login** page, the banner, the left-hand navigation pane or links frame, and images.

This chapter provides details on these tasks for Corporate Edition (CE), Standard Edition (SE), and Junk Mail Manager (JMM).

This chapter includes these topics:

- Corporate Edition Quick Branding
- Standard Edition/JMM Quick Branding
- Quick Branding Next Steps

For WebMail/WebCal Standard Edition and Junk Mail Manager only, you can Custom Brand (change any file). How to do so is discussed in Chapter 3, Custom Branding.

Read Chapter 1, Branding Basics, Tips, and Common Tasks for important information.

Corporate Edition Quick Branding

This section describes the five quick branding tasks available for Corporate Edition (CE), and the location of the Help files.

More information on style sheets is available in Working with the Cascading Style Sheets on page 25.

Branding the Style Sheets—CE

Corporate Edition style sheets are located in **extras/css** and **extras/css/themes** directories of the system brand zip.

The order in which the cascading style sheets are used is this:

- **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.

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**.

In Corporate Edition, there are seven default themes. You can change the look of these themes by editing the style sheet for each:

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

To also change the display names for the themes, modify these variables in the **mail/**locale.var file by changing the value on the right of the equal sign:

```
"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)";
```

For example, if you modify the system-5.css file and want to have the display name for it be "Summer" make this change in **mail/**locale.var:

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

Also change the name of the theme in it's style sheet; but do not change the name of the style sheet.

Branding the Login Page—CE

This section describes how to change the images and text used on the Corporate Edition **Login** page, shown in Figure 3.

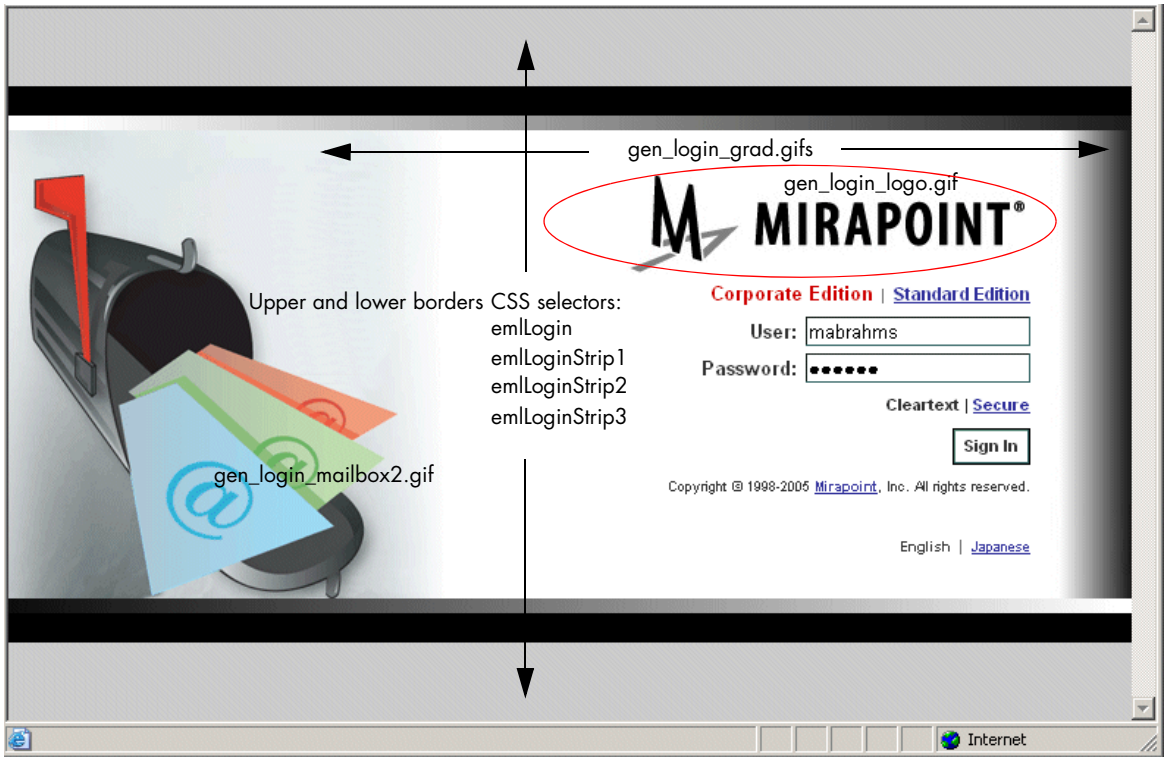


Figure 3 Corporate Edition Login Page Example (text variables explained next)

Corporate Edition Login Images

To change the logo in the **Login** page shown in Figure 3, replace these images found in the **images/eui** directory with images of your own; retain the names (you can ignore the "grey" versions of the GIFs, they are not used):

These set the grey fade on either side of the **Login** page shown in Figure 3:
gen_login_grad.gif and gen_login_grad.grey.gif
gen_login_gradient2.png and gen_login_gradient.png

This is the logo graphic "Mirapoint" and the trademark symbol:
gen_login_logo.gif (280x71 pixels)

This is the mailbox image:
gen_login_mailbox2.gif (304x319 pixels)

To modify the grey bands at top and bottom, the login table background color, modify these selectors in uui.css: "emlLogin" for the background color, "emlLoginStrip1," "emlLoginStrip2," and "emlLoginStrip3" for the black and gradient bands above and below.

Other images may be in the top-level **images/** directory.

When replacing the **Login** page logo, the optimal size in pixels of logos so that the logos do not appear distorted is 113 x 30 pixels; 72 DPI.

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 $variable provides the “Standard Edition” link, in a table. It uses the classes, “emlLoginTitle” (for “Corporate Edition”) and “emlLoginEdition” (for “Standard Edition”) for the text appearances; link styles use “A.emlLoginEdition” (for “Standard Edition”), defined in the uui.css style sheet.
- The **User** and **Password** option labels use the “emlLoginLabel,” “emlLoginInput,” “emlLoginBtn,” and “emlLoginError” classes defined in uui.css.
- "miravar_eml_versionfooter_login" = "$(eml_versionfooter_login)"; This variable provides the following text, in a table: “Copyright © 1998-2006 Mirapoint, 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 uui.css.
- "miravar_useothermode" = "$(useothermode)"; This variable provides the **Cleartext | Secure** mode options and uses the “emlLoginLink” and “A.emlLoginLink” classes defined in uui.css.
- You cannot change the locale footer, except through its css classes, “emlOtherLocale” and “A.emlOtherLocal” defined in uui.css.
- "miravar_left_indent" and "miravar_right_indent" = "$(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—CE

This section describes how to change the images and text used on the Corporate Edition banner, shown in Figure 4.

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

Figure 4 Corporate Edition Banner Example

When replacing the banner page logo, the optimal size in pixels of logos so that the logos do not appear distorted is 113 x 30 pixels; 72 DPI.

- 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, use the $(miravar_banner_content) variable in the **extras/**brandvars.var file.
- To change the "Welcome" message colors, modify the "genwelcome" class in the **extras/css/themes/**system-*.css files (you cannot alter the text, which is hard-coded as "Welcome *User Name*").
- To change the banner height, use the $(eui_bannderheight) variable in the **extras/**brandvars.var file. In this case, you must change the variable reference. 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 the Navigation Pane—CE

Use the “Navigation Pane Selectors” as described in the ce_usage.html file and called-out below, to change the colors of the tabs, link styles, and selected vs. non-selected items. See Figure 5.

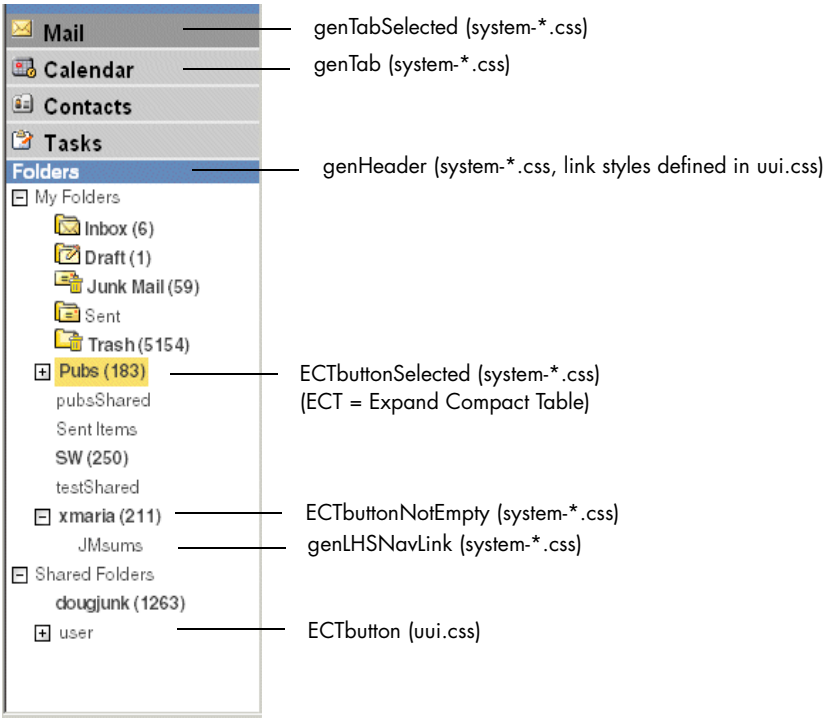


Figure 5 Corporate Edition Navigation Pane Example

Branding the Options Links—CE

You can remove or change any or all of the **Options** pop-up window left side links. This is particularly useful if you have a password schema and do not want users to be able to change their own password using the **Change Password** page. These variables (below) for the links have been added to **extras**/brandvars.var. You can remove any or all of the links of a branch by deleting the variables from brandvars.var; you cannot remove a branch entirely. You cannot re-name any of the branches or links

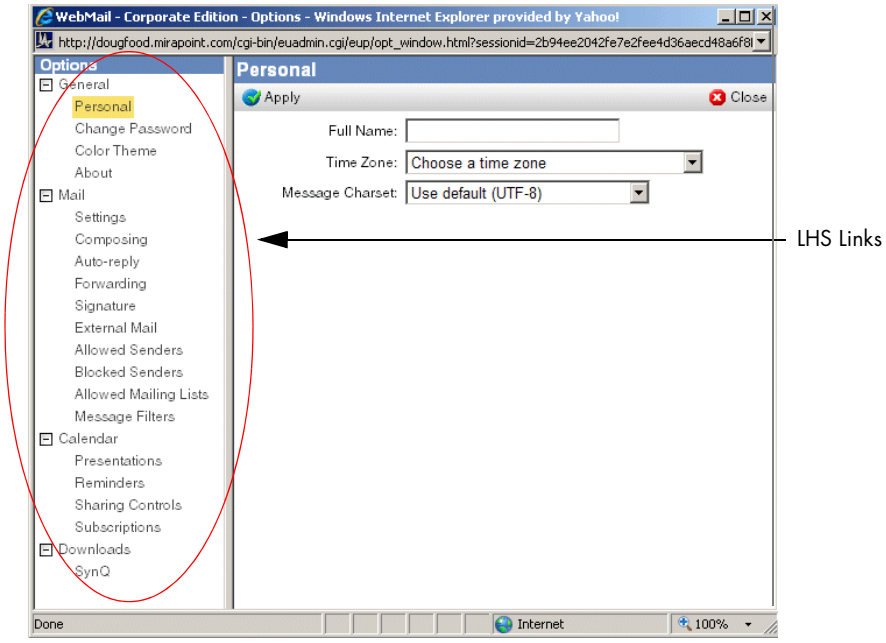


Figure 6 Corporate Edition Options Window

Delete these variables to remove that option:

- **General** branch, “miravar_options_general_prefererence_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)
  - $(subscribe_link)

- **Download** branch, “miravar_options_download_links”
  - $(downloads_link)

Branding the Help Files—CE

You brand Corporate Edition Help files by locating the HTML files and making changes; just be careful not to delete the paragraph tagging (e.g. “a name=wp1076530”) as they are used for cross-reference links.

Corporate Edition Help files use style sheets catalog.css and document.css, in the **Output** sub-directory of each of the two sets.

The Corporate Edition Help files are in these two directories in the system brand zip:

help/locale/*locale*/eml_I_Help/Output/: Help files for WebMail and WebCal Corporate Edition. Images are in a subdirectory of Output.

help/locale/*locale*/eml_W_Help/Output/: Help files for Corporate Edition WebMail only. Images are in a subdirectory of Output.

Standard Edition/JMM Quick Branding

This section describes the five quick branding tasks available for Standard Edition (SE) and Junk Mail Manager (JMM).

More information on style sheets is available in Working with the Cascading Style Sheets on page 25.

Detailed information on all the files that make up WebMail/WebCal SE and JMM, and additional available branding tasks, are provided in Chapter 3, Custom Branding.

Branding the Style Sheets—SE and JMM

The Standard Edition style sheet, style.css, is located in the **extras/css** directory of the system brand zip. Junk Mail Manager has an additional, cascading, style sheet **extras/css**/jmm_style.css. You do not need to brand this style sheet unless there are some things you want to look different in Junk Mail Manager.

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.

Branding the Login Page—SE and JMM

You can customize the Standard Edition WebMail, WebCal, or Junk Mail Manager login.html file with specified colors, a different logo, and links. This section describes changing the WebMail **Login** page defined in **mail**/login.html file. You can

follow these same procedures to change the WebCal **Login** page defined in **mcal/**login.html, or the Junk Mail Manager **Login** page defined in **spam/**rg_login.html.

There is an additional **Login** page file, login.html, in the **spam** directory that is used if your Junk Mail Manager is running on a Message Server. Typically, Junk Mail Manager runs on RazorGates, so you would use the rg_login.html file.

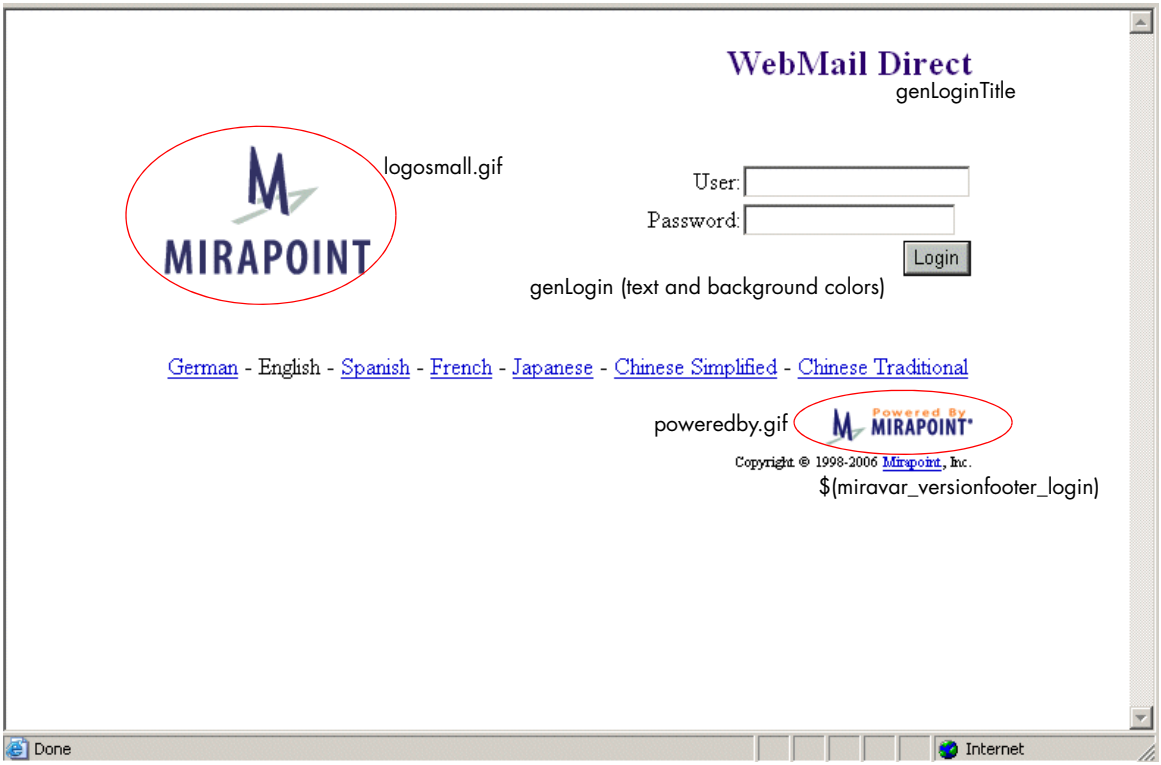


Figure 7 Standard Edition WebMail Direct Login Page Example

Standard Edition/Junk Mail Manager Login Images

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

Do not remove or replace the poweredby.gif; this logo should be left on the **Login** page of each. However, you do have some options as to which poweredby.gif you use, see Powered by Mirapoint on page 31 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/Junk Mail Manager Login Text

The “User” and “Password” text are defined in **mail**/login.html, **mcal**/login.html, and **spam**/rg_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, do this.

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—SE and JMM

This section describes how to change the images and text used on the Standard Edition banner, shown in Figure 8, defined in **mail**/banner.html. You can use this information to brand the WebCal banner defined in **mcal**/banner.html and the Junk Mail Manager banner defined in **spam**/banner.html.

Figure 8 Standard Edition Banner Example

Table 5 banner.html Variables

Element	Variable	Defined in…	Comments
BODY class	genNav	CSS and brandvars.var	Text and background colors
Apps Menu	$(appsmenu)	cell.var (key: appsmenu_tb)	❖ 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 there.
- 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:

```
<img src=”/wmi/your_imagfile.gif”>
```

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

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

```
<img src=”/wmi/your_imagfile.gif” hspace=10 vspace=20 width=100 height=50>
```

Branding the Links Frame—SE and JMM

The Links frame, **mail**/links.html, **mcal**/links.html, and **spam**/links.html, respectively, shown in Figure 9, displays on every WebMail 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 6 describes elements and variables. This frame is always used in Junk Mail Manager.

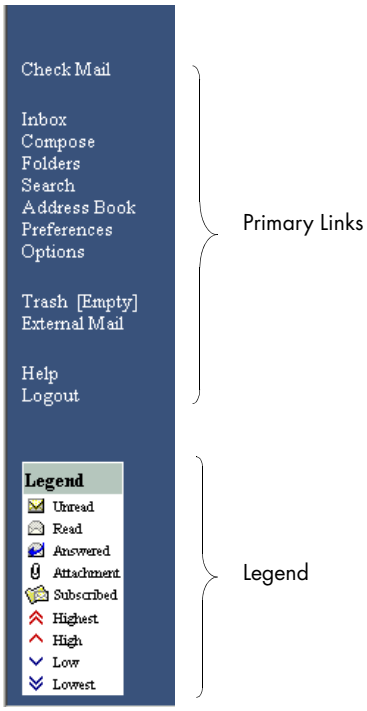


Figure 9 WebMail Standard Edition Links Frame Example

Table 6 links.html Variables

Element	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 and/or **spam**/links.html 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">
<img src="/wmi/your_ad_imagfile.gif"></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 and/or **spam**/links.html 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 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.

```
<img src="/wmi/your_button__imagename.gif">
```

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

Branding the Help Files—SE and JMM

Locate the HTML files and make whatever changes you want.

The Standard Edition and Junk Mail Manager Help 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*/spam/: Junk Mail Manager Help files.

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

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

Standard Edition and Junk Mail Manager Help files use the style.css and jmm_style.css style sheets, respectively, 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*/**.

Quick Branding Next Steps

Once you have branded the files, you must zip the files and publish them. How to do this is described in Viewing, Zipping, Publishing, and Assigning Your Brand on page 24. Additionally, you can access the online Help when using the Administration Suite to publish your brand, go to **Help > System > Branding**.

3 Custom Branding

Custom branding consists of changing any of the files that make up the Standard Edition (SE) "five end-user applications:" WebMail, WebCal, Options, Spelling, and Address Book; and Junk Mail Manager (JMM). These files are not available for branding Corporate Edition.

This chapter provides a reference of the files available for custom branding and some application-specific branding tasks. For information on quick branding SE and JMM, see Chapter 2, Quick Branding.

This chapter includes these topics:

- 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
- Branding the Junk Mail Manager Pages
- Custom Branding 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.

Read Chapter 1, Branding Basics, Tips, and Common Tasks for important information.

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 90.

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:

- **No Frames**: The pages display without the use of frames.
- **Frames**: The pages display in three frames; a "banner," a left-side "links," and the page "template" frame.
- **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 at the same time.

Users select a mode in the **Preferences** page of WebMail or WebCal. The default mode is **Frames**; you can change the default with branding. For users to see only this mode, you must comment out the other modes on the **Preferences** page (prefs.html). If you want users to have the ability to see branding on all three modes, you must brand all three, which involves branding additional files.

Frames with JavaScript Mode

Figure 10 is an example of the **Checkmail** page in the **Frames with JavaScript** mode, with screen elements called-out. 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, which in the example shown in Figure 10, displays the message contents and commands; read.html when a message is selected and read_blank.html when no message is selected.

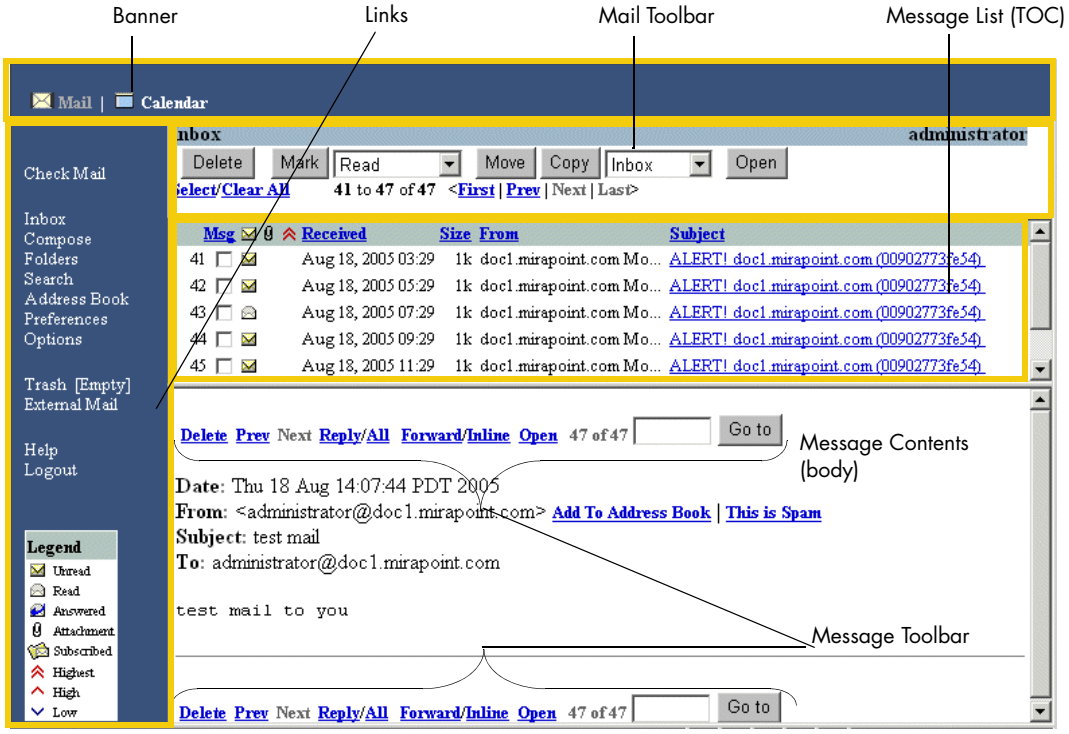
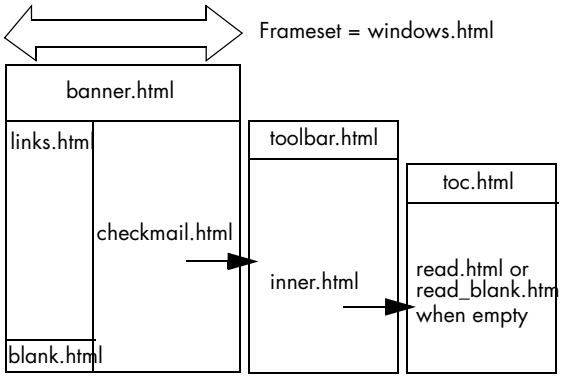


Figure 10 Frames With JavaScript Mode, WebMail Standard Edition

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. The **Frames** mode works much the same way, only there is no bottom frame to display messages. Instead, your message (read_s.html) displays in the frame that displayed your message list (toc.html).

Table 7 depicts the **Frames with JavaScript** mode.

Table 7 Frames with JavaScript Mode Example

Description	Example
The frameset is defined in windows.html. The top frame is created by banner.html; links.html and blank.html create the left frame. This mode sometimes uses the same files as the **No Frames** mode. You can add JavaScript to all three modes if you want, but the **Frames with JavaScript** mode requires it for certain functions.	

Frames Mode

Figure 11 is an example of the **Checkmail** page in the **Frames** mode, the screen elements are called-out. 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).

For the **Frames** mode, the **checkmail** frame references checkmail_s.html, not checkmail.html (the frameset definition file that the **Frames with JavaScript** mode gets). For the **Frames** mode, the **TOC** elements are defined in checkmail_s.html. When you click on a message link, that message (read.html) replaces the **checkmail** frame (checkmail_s.html).

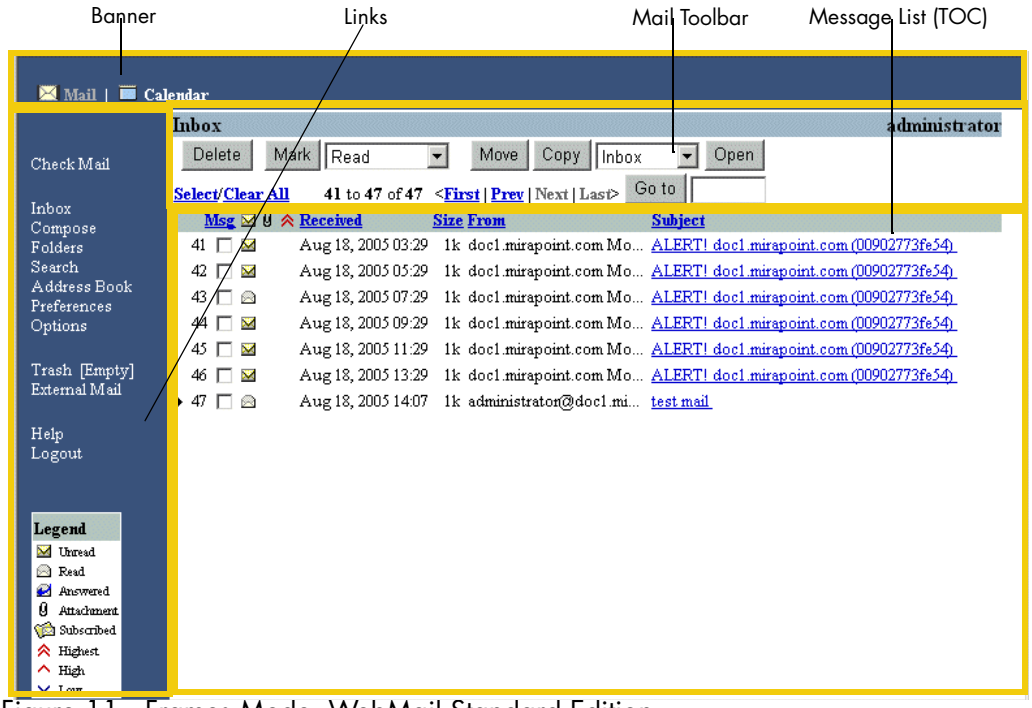
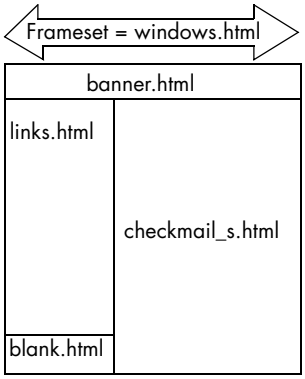


Figure 11 Frames Mode, WebMail Standard Edition

Table 8 depicts the **Frames** mode.

Table 8 Frames Mode Example

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

No Frames Mode

In WebMail, the **No Frames** mode has several files in common with the **Frames** mode, they are:

- checkmail_s.html, shown in Figure 12: Displays the Inbox or selected folder and performs a mail check
- search_s.html: Displays the **Search** page
- read_s.html: Displays your message

The two modes 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. But you must brand those common files in order for the **Frames** mode, typically the default mode, to be branded. Figure 11 is an

example of the **Checkmail** page in the **No Frames** mode. When you click on a message link, that 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 "simplelinks"

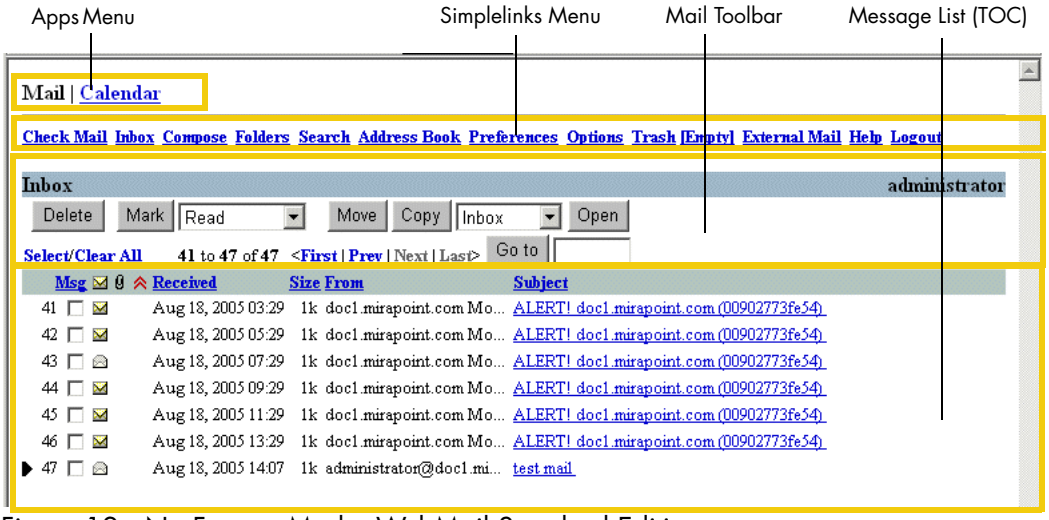


Figure 12 No Frames Mode, WebMail Standard Edition

Table 9 depicts **No Frames** mode.

Table 9 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.	checkmail_s.html

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, shown in Figure 13, 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 10 describes elements and variables.

Figure 13 Banner, Calendar highlighted

Table 10 banner.html Variables

Element	Variable	Defined in...	Comments
BODY style	genNav	CSS and brandvars.var	Text and background colors

Table 10 banner.html Variables (Continued)

Element	Variable	Defined in...	Comments
Apps Menu	$(appsmenu)	cell.var (key: appsmenu_tb)	❖ 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 14, 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 11 describes elements and variables.

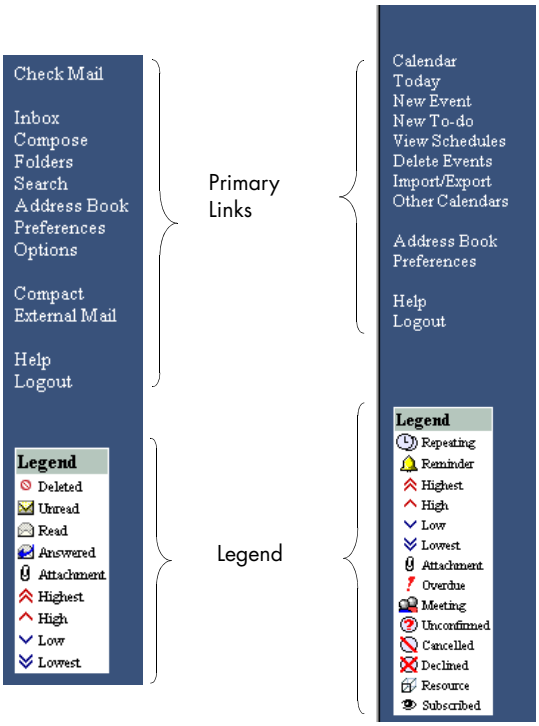


Figure 14 Links Frame, WebMail on the left, WebCal on the right

Table 11 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	❖ <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.
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 15 with the **Delete to Trash** feature disabled (default) and in Figure 16 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 12 describes elements and variables.

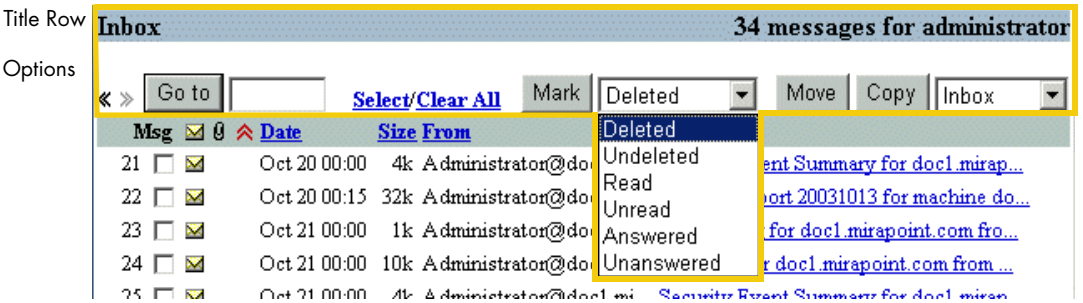


Figure 15 toolbar.html Frames with JavaScript Mode, Delete to Trash *disabled*

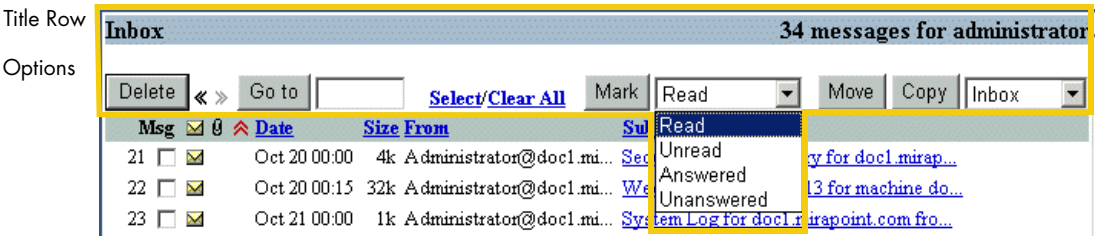


Figure 16 toolbar.html Frames with JavaScript Mode, Delete to Trash *enabled*

Table 12 toolbar.html Variables

Element	Variable	Defined in...	Comments
BODY style	$(miravar_body)	CSS and brandvars.var	❖ 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)	❖ 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

Table 12 toolbar.html Variables (Continued)

Element	Variable	Defined in...	Comments
Approve button	$(qtnapprovebuttonjs)	cell.var (key: qtnapprovebuttonjs)	❖ Adds/removes **Approve** button ❖ Change position in template; change text in cell.var
Rescan button	$(virusscanbuttonjs)	cell.var (key: virusscanbuttonjs)	❖ Adds/removes **Rescan** button ❖ Change position in template; change text in cell.var
Delete button	$(deltrashbuttonjs)	cell.var (key: deltrashbuttonjs)	❖ Adds/removes **Delete** button ❖ Change position in template; change text in cell.var
Page links	$(prevplink)	cell.var (keys: prevpjs_on, prevp_off)	Change position in template; change text in cell.var
	$(nextplink)	cell.var (keys: nextpjs_on, nextp_off)	
Select/ Clear All links	N/A	template	❖ Uses JavaScript ❖ Change position or text in template
Mark options	$(markdeletedjs)	cell.var (system keys)	❖ 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)		
	$(usetrash)		Defined by the user **Delete to Trash** preferences setting; adds/removes **Deleted** and **Undeleted** options
Move and **Copy** options	$(mboxlistopt)	cell.var (system key)	❖ Folder drop down list ❖ Change position or text in template

toc.html (WebMail only)

The Table of Contents frame, toc.html, shown in Figure 17, 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 13 describes elements and variables.

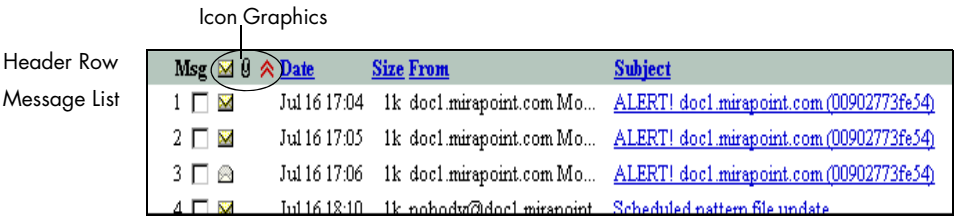


Figure 17 toc.html Frames with JavaScript Mode

Table 13 toc.html Variables

Element	Variable	Defined in...	Comments
BODY style	$(miravar_body)	CSS and brandvars.var	❖ 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	❖ 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
Sorted on icon (^ and ∨)	$(msgorderimage) $(seenorderimage) $(attachorderimage) $(priorityorderimage) $(dateorderimage) $(sizeorderimage)	system defined	Display the Ascending/Descending **Sorted on** icons (^/∨)
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)	❖ 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 25 for details on the CSS.

WebMail Frameset Definition Files

Table 14 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 14 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).
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 15 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 15 WebMail (mail/) Process Files

File Name	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 16.

Table 16 Other vartab Files used by the Applications

Directory	File	Description
mail	charsets.var	Character set name definitions for the **Message Charsets** list used by the **Preferences** and **Compose** pages (prefs.html and compose.html). The vartab key is the actual string that gets inserted into the mail message header (e.g., charset="UTF-8"). 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 (prefs.html). 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 93 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 locale.var 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 83.

WebMail Standard Edition HTML Template Files Per Mode

Table 17 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 17 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.

Table 17 WebMail (mail) HTML Template Files (Continued)

Template File	NF	F	F+ JS	Description
nocookie.html	✔	✔	✔	Cookies must be enabled page.
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.
readreciept.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.

Sometimes stopping and re-starting WebMail and WebCal is needed 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 18 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 18 Default Preference Keys Explained

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 18) 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 Austrailia|Brisbane”;
charsetdefault = “ISO-2022-JP”;
draftfolderdefault = “Draft”;
trashfolderdefault = “Trash”;
junkfolderdefault = “Junk”;
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 19.

Table 19 ANSI C Function strftime Format Directives

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

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 chapter 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) Standard Edition. Groupcal is a super-set of WebCal; you must have both licenses in order to get the Groupcal functions.

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 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.

This book describes the files for Groupcal Direct, WebCal Direct being 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 11 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 11 shows what the Calendar **Preferences** page **User Information** view in **Frames** mode looks like.

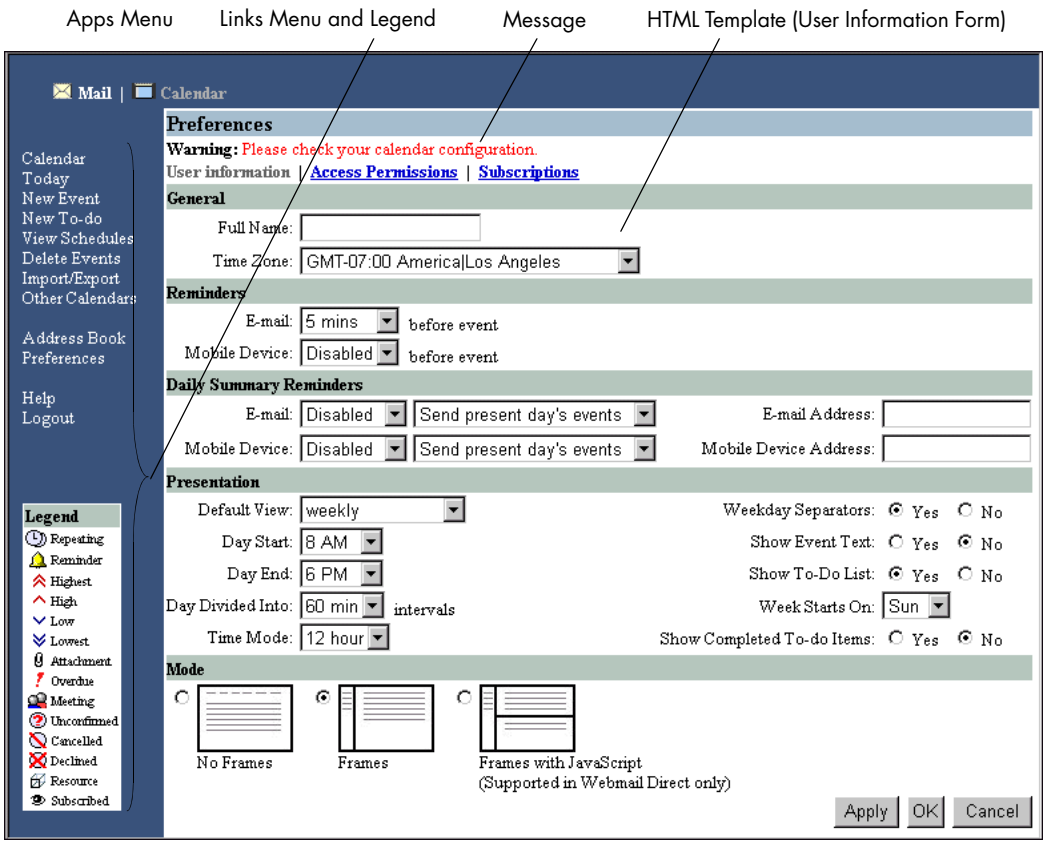


Figure 18 WebCal Screen Elements—Frames Mode

No Frames Mode

Figure 19 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 11 shows what the Calendar **Preferences** page **User Information** view in **No Frames** mode looks like.

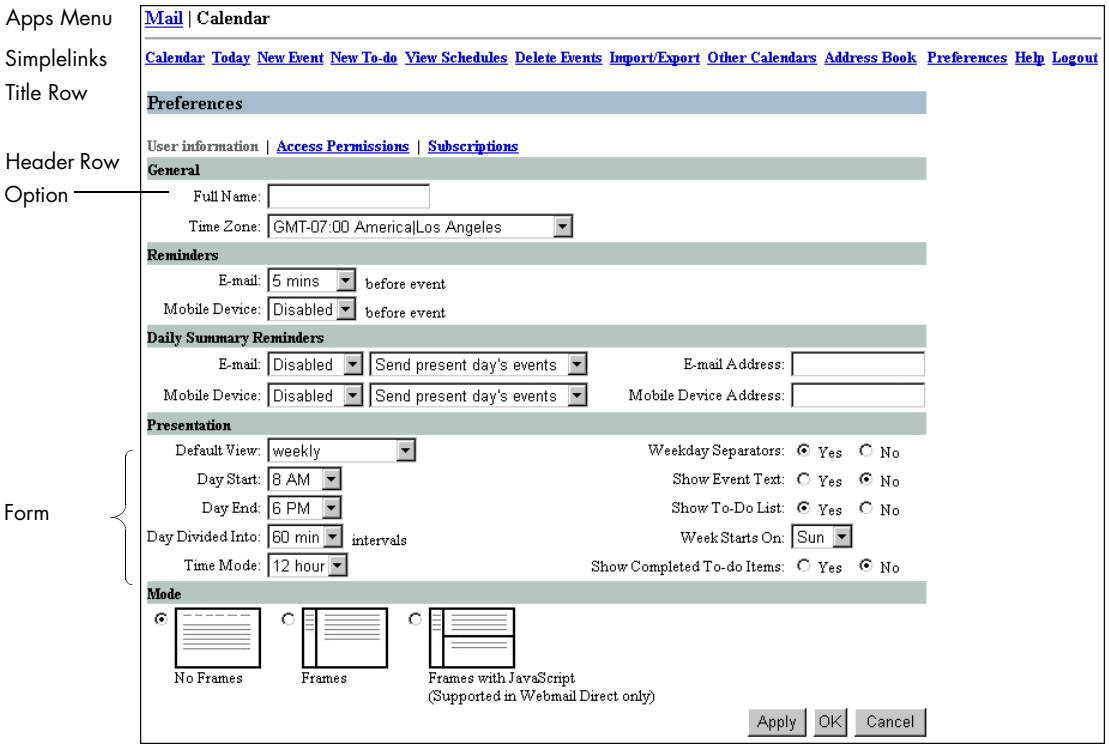


Figure 19 WebCal Screen Elements—No Frames Mode

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 25 for details on the CSS.

WebCal Frameset Definition Files

Table 20 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 20 WebCal Frameset Definition Files

Template File	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 21 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 21 WebCal Process (mcal/) Files

File Name	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 17 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 22 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.

Table 22 WebCal (mcal/) HTML Template Files (Continued)

Template File	Description
confirmprefs.html	Confirmation page for preferences requests.
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.
nenav.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.
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.

Table 22 WebCal (mcal/) HTML Template Files (Continued)

Template File	Description
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.
todays.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 mcal/locale/*localename*/cell.var. Unlike WebMail, you do not need to append "default" to the key; instead, you simply change the value. Table 23 explains a few common ones.

Table 23 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	

In addition, there are many more display factors and options that can be set in this file.

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 24. In addition, see mcaldmsg.var for other variable definitions.

Table 24 Email Messages Defined in mcaldmsg.var

Variable	Default Message	Description
invitationnew	"You are invited to attend the following meeting: Title: $(synopsis) Owner: $(owner) Date: $(eventdate) Time: $(eventtime) Participants: $(attendees) $(location_section) Description: $(description)"	❖ Sent to **Meeting Attendees** when a new meeting is created and **Send E-mail** is selected ❖ These variables are defined by user input
external_invitationnew	"You are invited to attend the following meeting: Title: $(synopsis) Owner: $(owner) Date: $(eventdate) Time: $(eventtime) To confirm or decline attendance, please visit the following web page: $(externalurl) $(location_section) Description: $(description)"	❖ 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 24 Email Messages Defined in mcaldmsg.var

Variable	Default Message	Description
invitationsubject	"$(owner): $(synopsis)"; external_invitation= "The following meeting has been modified: Title: $(synopsis) Owner: $(owner) Date: $(eventdate) Time: $(eventtime) To confirm or decline attendance, please visit the following web page: $(externalurl) $(location_section) Description: $(description)"	❖ Sent to **Meeting Attendees** when a meeting is modified by the meeting owner ❖ These variables are defined by user input
external_invitationsubject	"$(owner): $(synopsis)"; external_invitation= "The following meeting has been modified: Title: $(synopsis) Owner: $(owner) Date: $(eventdate) Time: $(eventtime) To confirm or decline attendance, please visit the following web page: $(externalurl) $(location_section) Description: $(description)"	❖ Sent to **External Meeting Attendees** when a meeting is modified by the meeting owner ❖ These variables are defined by user input

Table 24 Email Messages Defined in mcaldmsg.var

Variable	Default Message	Description
acceptsubject	"meeting accepted by $(attendee)"; accept = "$(attendee) has accepted your invitation to: Title: $(synopsis) Owner: $(owner) Date: $(eventdate) Time: $(eventtime) Participants: $(attendees)"	❖ Sent to the meeting owner when **Accept** is clicked by an invited meeting attendee ❖ These variables are defined by user input
declinesubject	"meeting declined by $(attendee)"; decline = "$(attendee) has declined your invitation to: Title: $(synopsis) Owner: $(owner) Date: $(eventdate) Time: $(eventtime) Participants: $(attendees)"	❖ Sent to the meeting owner when **Decline** is clicked by an invited meeting attendee ❖ These variables are defined by user input
cancellationsubject	"$(owner): $(synopsis)"; cancellation= "This meeting has been cancelled by the owner: Title: $(synopsis) Owner: $(owner) Date: $(eventdate) Time: $(eventtime) Description: $(description)"	❖ Sent to **Meeting Attendees** when a new meeting is created and then canceled by the meeting owner ❖ These variables are defined by user input
external_cancellationsubject	"$(owner): $(synopsis)"; external_cancellation= "This meeting has been cancelled by the owner: Title: $(synopsis) Owner: $(owner) Date: $(eventdate) Time: $(eventtime) Description: $(description)"	❖ 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 24 Email Messages Defined in mcaldmsg.var

Variable	Default Message	Description
dailysummary	"Summary of events for $(date): Time Title ---- ----- $(eventlist)"	❖ 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)"	❖ 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 25 Options/Account (user/) Process Files

File Name	Description and Dependencies
cell.var	Contains variable definitions.
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 26 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 26 Options/Account (user/) HTML Template Files

File Name	Description
antispamfilter.html	Creates the **Junk Mail Control** page **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.
chgpass.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.

Table 26 Options/Account (user/) HTML Template Files

File Name	Description
login.html	Used by the Administration Suite.
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 27 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 28 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 20 shows the Address Book Simplelinks menu for WebMail; Figure 21 shows the Address Book Simplelinks menu for WebCal.

Figure 20 Address Book Simplelinks Menu for WebMail Direct

Figure 21 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 29 Address Book (addrbook/) Process Files

File Name	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 93 for details.
locale.var	Contains customizable system and error messages.

Address Book HTML Template Files

Table 30 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 30 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.

Table 30 Address Book (addrbook/) HTML Template Files (Continued)

File Name	Description
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**.
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",                "Department",
"Job Title",                 "Job Title",
"Business Street",           "Business Street",
"Business Street 2",         "Business Street 2",
"Business Street 3",         "Business Street 3",
"Business City",             "Business City",
"Business State",            "Business State",
"Business Postal Code",      "Business Postal Code",
"Business Country",          "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 31. **Note:** "mcal" indicates a variable used by Calendar Address Book.

Table 31 Contact Table Row Format Variables

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

The fields of the current contact are defined through the following variables described in Table 32.

Table 32 Contact Table Row Data Variables

Variable	Description
categoryIndex	The index of the category to which this contact belongs.
categoryName	The localized name of the category to which this contact belongs.

Table 32 Contact Table Row Data Variables (Continued)

Variable	Description
primarynumber	The primary phone number for the contact.
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> </b></font></td>
    <td><font size=-1> </font></td>
    <td align=right nowrap><font size=-1><b> </b></font></td>
    <td><font size=-1> </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.

Branding the Junk Mail Manager Pages

This chapter provides an overview of the files in the **spam**/locale/*localename*/ and **spamu**/locale/*localename*/ directories that define the pages for Junk Mail Manager.

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.

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 14 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 33 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.
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. You can make global changes by changing variable definitions in these files.

Table 34 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 35 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. You can change element terminology, colors (in some cases), add graphics, and reorder data fields using these files.

Table 36 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.
confirmdel.html	Creates the **Spelling** page that opens when a user clicks **Spelling** when in the **Compose** window.
confirmwindow.html	Confirmation page for preference settings requests.
home.html	Creates the opening page.
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.
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.

Table 36 Junk Mail Manager (spam/) HTML Template Files

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

Table 37 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.

Custom Branding Next Steps

Once you have branded the files, you must zip the files and publish them. How to do this is described in Viewing, Zipping, Publishing, and Assigning Your Brand on page 24. Additionally, you can access the online Help when using the Administration Suite to publish your brand, go to **Help > System > Branding**.

Index

A

adding
- a linked advertisement, Standard Edition & JMM 50
- images 49
- links to Corporate Edition banner 43

Address Book pages
- about 90
- adding LDAP search fields 96
- branding 90
- branding options 93
- contacts variables 94
- localized import/export 93
- process files 91
- template files 91
- WebMail vs. WebCal 90

attachments, setting in WebMail 75

B

banner
- Corporate Edition 43
- Standard Edition & JMM 48

banner.html 60

branding
- assigning domains 25
- configuring local hosts 23
- creating a domain sensitive login page 36
- creating delegated domains 23
- custom brand document 22
- custom vs. quick 17
- downloading the factory zip file 24
- error messages 33
- getting ready 21
- Help files 46
- named brands, about 20
- process 21
- prohibited practices 20
- publishing your brand 24
- replacing the login logo image 42
- reverting back to the default 34
- setting up your workstation 22
- two types of variables 28
- using the usage.html file 27
- variables format 30
- variables, published, run, rendered 29
- working with CSS 25
- working with variables 28

branding files
- cell.var, about 67
- HTML templates 72
- images 31
- locale.var, about 68

branding, Address Book
- about 90
- localized import/export 93
- options 93

branding, Corporate Edition
- banner 43
- login page 41
- navigation pane 44
- options window 44
- quick branding 39
- style sheets 39

branding, Junk Mail Manager
- process files 98
- template files 99

branding, Options/Account pages
- about 87
- process files 88
- template files 88

branding, Spelling pages
- about 89
- process files 89
- template files 90

branding, Standard Edition & JMM
- adding a linked advertisement 50
- adding images 49
- banner 48
- custom branding 53
- links.html 49

login page 46
style sheets 46
substituting buttons for links 51
branding, WebCal
about 76
frameset files 80
options 83
process files 80
template files 80
branding, WebMail
frameset definition files 66
options 73
process files 66
template files 70
brands
assigning domains to 25
named, about 20
publishing 24
two types 20
updating 34
brandvars.var
about CSS fallback 68
CSS fallback plan 27

C

cell.var, about 67
charset
setting in WebMail 73
WebMail, charsets.var 66
common resource names 30
configuring
domains 23
workstation host file 23
Corporate Edition
banner 43
branding the Help pages 46
branding the themes 40
custom branding vs. quick branding 17
login page 41
login page images 41
login page text 42
navigation pane 44
options window 44
quick branding 39
replacing the banner logo image 43
style sheets 39
upgrading to 35
creating
delegated domains for branding 23
domain sensitive login pages 36
variables 29
CSS
about usage.html 27
fallback plan 27
terminology 26
working with 25
custom brand document, about 22
custom branding vs. quick branding 17
custom branding, about 17

D

date format, setting in WebMail 74
default factory brand, reverting back to 34
defaults
mode settings, WebCal 83
mode settings, WebMail 73
reverting to the factory brand 34
setting the date format, WebMail 74
WebMail attachment handling 75
WebMail charset definition 73
display_charset values 32
domain sensitive login page 36
domains
assigning to brands 25
creating for branding 23
downloading
factory brand zip file 24
Mirapoint powered-by logo 31

E

email messages, branding for WebCal 83
error messages, definitions 33

F

factory brand, reverting back to 34
fallback plan for the CSS 27
files, directory structure 18
Frames mode
definition 58
WebMail 57
Frames with JavaScript mode
definition 56
WebMail 55
frameset definition files
WebCal 80
WebMail 66

H

Help pages, branding 46

I

images
branding 31

replacing the banner, CE 43
replacing the login logo, CE 42

J

JavaScript files, **extras directory** 67
Junk Mail Manager
about 97
brand.vars, about CSS fallback 68
process files 98
template files 99
what can be branded 54

L

LDAP, adding search fields 96
links, adding to Corporate Edition banner 43
links.html 61
links.html, Standard Edition & JMM 49
locale.var
about 68
definition 18
localized import/export, Address Book 93
logging in
to Mirapoint Support 13
login pages
branding, Corporate Edition 41
images, Corporate Edition 41
images, Standard Edition & JMM 47
making domain sensitive 36
Mirapoint logo 31
text, Corporate Edition 42
text, Standard Edition & JMM 48

M

Mirapoint
logo 31
Support Site URL 13
modes
selecting 55
setting defaults, WebCal 83
setting defaults, WebMail 73
WebCal pages 77
WebMail template files 70
WebMail/WebCal 55
MTL, definition 13

N

named brands, about 20
navigation pane, Corporate Edition 44
No Frames mode
definition 59
WebMail 59

O

options window, Corporate Edition 44
Options/Account pages
about 87
branding 87
process files 88
template files 88

P

planning a custom brand 54
process files
Address Book pages 91
definition 18
Junk Mail Manager pages 98
Options/Account pages 88
Spelling pages 89
WebCal pages 80
WebMail 66
publishing
start/stop WebMail and WebCal 24
publishing your brand 24
publish-time variables, definition 28

Q

quick branding, about 17

R

reverting back to the default factory brand 34
run-time variables, definition 29

S

screen elements
custom branding 54
WebCal pages 79
WebMail 59
selecting a display mode 55
setting up branding environment 22
Spelling pages
branding 89
process files 89
template files 90
Standard Edition
contacts variables 94
custom branding 53
custom branding vs. quick branding 17
files and directories 18
what can be branded 54
Standard Edition & JMM
adding a linked advertisement 50
banner 48
brand.vars, about CSS fallback 68

branding the Help files 46
branding the style sheets 46
links.html 49
login page images 47
login page text 48
login page, quick branding 46
substituting buttons for links 51
style sheets
Corporate Edition 39
Standard Edition & JMM 46
substituting buttons for links, Standard Edition & JMM 51
support
getting a login ID 13
getting a Mirapoint Support login ID 13
system variables, about 30

T

template files
Address Book pages 91
Junk Mail Manager pages 99
Options/Account pages 88
Spelling pages 90
WebCal pages 80
template HTML files
branding 72
definition 18
themes, branding 40
Three-Way Merge, branding upgrade technique 35
toc.html 64
toolbar.html 62
troubleshooting
about quotes 31
Administration Login page not displaying 34
brand publishing problem 24
error messages 33
getting a Mirapoint Support login ID 13
Help files URL 51
URLs vs. Forms 32
typographic conventions 14

U

upgrading
to Corporate Edition 35
your brand 34
user tasks
logging in 21
selecting a display mode 55

V

variables
about 28
about quotes 31
format 30
published, run, rendered 29
system, about 30
two types 28
vartab files
definition 18
per application 69

W

WebCal pages
about 76
branding 76
branding email messages 83
branding options 83
frameset definition files 80
how they are defined 79
modes 77
process files 80
screen elements 79
setting default mode 83
template files 80
WebMail
branding options 73
frameset definition files 66
how the pages are defined 66
process files 66
screen elements 59
setting attachment handling 75
setting default mode 73
setting the charset 73
setting the date format 74
template files per mode 70
upgrading to Corporate Edition 35
WebMail/WebCal modes 55

Z

zip
directory structure 18
getting ready to publish 24