

Sun Java System Communications Express 6.3 Customization Guide



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-4441-11
March 2009

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and SunTM Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certains composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

Preface	13
1 Introduction to Communications Express and Customization	17
Components of Sun Java System Communications Express 6.3	17
2 Customizing General Features in Communications Express	19
Guidelines for Customization	19
Customization Do's and Don'ts	20
List of Customizable and Configurable Files for Calendar and Address Book	21
Images	22
Text	23
▼ To Customize Text for a Domain	23
Themes	24
▼ To Create a new Theme	24
▼ To Modify a Skin	24
Login Screen	25
Replacing the Logo With a Custom Graphic	25
Example: Logo Modification	26
Changing the Title Text	26
Banner	27
The Application Bar	27
3 Customizing General Features in Calendar	35
Icons	35
The Order of Calendar Toolbar Icons	36
Icons in Various Calendar Views	41
The Toolbar Images	42

Events/Task Related Informative Icons	43
The Error Icon for Full Page Errors	43
Icons in Pop ups	44
Search for Events	44
Views	45
Tasks Search View	45
The Task Column Order in Tasks Printable View	46
Tasks Error View	49
Events view	49
Tasks View	50
Pop ups	50
Full Page Error	51
Customizing the Layout for Full Page Errors	51
 4 Customizing General Features in Mail	53
Customization Scenarios	53
Customizing the Skin to Modify Global Look and Feel	53
Customizing Text	53
Customizing Mail by Domain	54
Customizing Mail Icons	54
Adding Extra Features	54
Understanding the File Layout in Mail	54
Understanding Basic Interfaces in Mail	55
Inbox	55
Message Display	56
Folder Management	56
Compose Window	57
Message Search Window	58
 5 Customizing User Interface Features in Mail	59
The Mailbox Tool Bar	59
Customizing the Mailbox Tool Bar	60
Example: Mailbox Tool Bar Modifications	60
Message Listing	61
Modifying Message List Window	61

Example: Message Listing Modifications	62
Customizing The Message Display Window	62
Modifying Message Display Window	63
Modifying the Message Display Window to Display User Defined Header Fields	63
The Message Tool Bar	64
Modifying the Message Tool Bar	64
The Message Composition Window	65
Modifying the Message Composition Window	65
Aligning the User Interface from Right to Left (For Arabic only)	66
▼ To Align the User Interface From Right to Left	66
6 Customizing Advanced Features in Mail	67
Files for Customizing the Mail User Interface	67
Attachment Options	68
Collect Mail from Another Server	68
Message Composition	69
Mailbox Management	69
Return Receipt	69
Understanding Shared Folders	70
▼ To Share Folders	70
Default LDAP Attributes for Users	70
7 Customizing General Features in Address Book	73
Address Book Icons and Labels	73
Customizing Icons	74
Style Sheets	75
What can be Customized in Style Sheets	76
The Main Search Page	78
Search Result Table	79
New/Edit Contact Window	79
Phone	80
Email	83
Dates	84
Web Addresses	86
IM Nicknames	86

View Contact Window	88
New Group or Edit Group Window	89
Group Details	89
The View Group Window	90
Printable Window	90
The Import/Export Address Book Window	90
8 Customizing Domains	93
Default User Preferences	93
▼ To Customize the Default User Preferences for a Domain	93
Themes for a Specific Domain and Locale	94
▼ To Customize a Theme for a Specific Domain	94
Mail Customization for a Specific Domain	95
Domain Customization Based on URL	96
Address Book Customization for a Specific Domain and a Locale	96
▼ To Customize Address Book for a Specific Domain and Locale	96
9 Localizing Communications Express	99
Calendar and Options	99
Date/Time Formats for Calendar	99
Languages that use Characters Outside the ASCII Range	101
Mail	101
Specific Locales	101
Adding Locales	102
Changing Source Encoding	103
▼ To Enable Character Encoding	103
Address Book	104
Customizing Address Book for a Specific Locale	104
Index	105

Figures

FIGURE 2-1	Communications Express Login Screen.	25
FIGURE 2-2	Sun Logo and Link	26
FIGURE 2-3	Example Of a Corner Logo	26
FIGURE 2-4	Login Screen with Customized Logo	26
FIGURE 2-5	Communications Express Title Text	26
FIGURE 2-6	Left Banner Image Before Customization	27
FIGURE 2-7	Customized Left Banner Image	27
FIGURE 2-8	Right Banner Image Before Customization	27
FIGURE 2-9	Customized Right Banner Image	27
FIGURE 2-10	Default application tab order	28
FIGURE 2-11	Changing the Application Tab Order to Display Address Book, Mail, Calendar and Options	31
FIGURE 3-1	Order of Calendar Toolbar Icons	36
FIGURE 4-1	Communications Express Inbox Screen	56
FIGURE 4-2	Message Display	56
FIGURE 4-3	Folder Management Screen	57
FIGURE 4-4	Compose Window	57
FIGURE 4-5	Message Search Window	58
FIGURE 5-1	Mailbox Tool Bar	59
FIGURE 5-2	Message List Window	61
FIGURE 5-3	Message Display Window	62
FIGURE 5-4	Message Tool Bar	64
FIGURE 5-5	Compose Window	65
FIGURE 7-1	Customizing Phone	81
FIGURE 7-2	Customizing Email	84
FIGURE 7-3	Customizing the Import/Export Address Book Window.	91

Tables

TABLE 2-1	List of files that often contain bug fixes	20
TABLE 2-2	List of files that should be edited with caution while customizing Communications Express	20
TABLE 2-3	List of files related to global fonts and colors	21
TABLE 2-4	List of Customizable files for Calendar and Address Book	21
TABLE 2-5	Location of the Common images in < <i>uwv-images-dir</i> >	22
TABLE 3-1	Customizing Icons	42
TABLE 3-2	Customizing the Calendar Toolbar Icons	42
TABLE 3-3	Customizing Task Related Informative Icons	43
TABLE 3-4	Customizing Error Icon for Full Page Errors	44
TABLE 3-5	Customizing Icons in Pop ups	44
TABLE 3-6	Customizing Pop ups	50
TABLE 4-1	JavaScript Files that are Used in Mail	55
TABLE 6-1	Communications Express User Interface Customizable Features	67
TABLE 7-1	Commonly used templates in <i>search-template.xml</i> and <i>search.xml</i>	78
TABLE 7-2	Templates Applicable for New/Edit Contact Window	79
TABLE 7-3	Templates Defined for Various Sections in the New/Edit group.	89
TABLE 8-1	Directory Structure for the Domain <i>siroe.com</i>	95
TABLE 8-2	Linking Multiple Domains to Few Distinct Brands	96
TABLE 9-1	Localizing Date/Time Formats for Calendar	100
TABLE 9-2	Localization Rules	100
TABLE 9-3	Communications Express Specific Locales	102

Examples

EXAMPLE 2-1	Code for Customizing the Application Tab Order	28
EXAMPLE 2-2	Customizing the order of the application tabs	32
EXAMPLE 3-1	Default Code that Displays the Calendar Toolbar Icons	36
EXAMPLE 3-2	Code After Modifying the Resource Bundle	45
EXAMPLE 3-3	Default Code that Displays the Table Column Order	46
EXAMPLE 5-1	Altered Tool Bar Layout (mbox_fs_lr.jsp)	60
EXAMPLE 5-2	Altering Tool Bar Text (en/i18n.js)	61
EXAMPLE 5-3	Altering List Window Text (en/i18n.js)	62
EXAMPLE 5-4	Changes made to the msg_fs_lr.jsp file	63
EXAMPLE 5-5	Altering Message Tool Bar Text (en/i18n.js)	64
EXAMPLE 5-6	Altering Composition Window Text (en/i18n.js)	66
EXAMPLE 5-7	Aligning the User Interface from Right to Left	66
EXAMPLE 6-1	Customizable LDAP attributes	70
EXAMPLE 6-2	Display of Banner Advertisements Depending on the User Class:	71
EXAMPLE 7-1	Changing Icon	74
EXAMPLE 7-2	Customization of the Label	75
EXAMPLE 7-3	Code Before Modification of the Default Definition in Dictionary-en-xml	75
EXAMPLE 7-4	Customized Default Definition in Dictionary-en-xml	75
EXAMPLE 7-5	Class attribute of HTML tags	76
EXAMPLE 7-6	Class attribute of XSL tags	76
EXAMPLE 7-7	Before Changing the Definition of a Primary Button Display "Btnl'	77
EXAMPLE 7-8	Class Used in the XSL File for the "Close' Button	77
EXAMPLE 7-9	After Changing the Definition of a Primary Button Display "Btnl'	77
EXAMPLE 7-10	After Changing Class Attribute Reference for the Desired HTML Tag	77
EXAMPLE 7-11	Code that Displays Address Before Online Information	80
EXAMPLE 7-12	Code that Displays Online Information Before Address	80
EXAMPLE 7-13	Default Code that Displays Work Phone as Priority 1 and Home Phone as Priority 2	81
EXAMPLE 7-14	Code that Displays Home Phone as Priority 1 and Work Phoneas Priority 2	82

EXAMPLE 7-15	Default Code that Shows Horizontally Aligned Fields and Displays Birthday as the First Field and Anniversary as the Second	84
EXAMPLE 7-16	Code that Displays the Field in a Vertical Layout and also Rearranges Anniversary as the First Date and Birthday as Second	85
EXAMPLE 7-17	Default Code that Displays SunONE as the Default Service	86
EXAMPLE 7-18	Modified Code that Displays Yahoo as the Default Service	87
EXAMPLE 7-19	Code to be Used for Adding the Email Address that is Supported for a Group	89
EXAMPLE 7-20	Code Before Modifying the Options List	91
EXAMPLE 7-21	Code After Modifying the Options List	92
EXAMPLE 9-1	Modifying the charset in the i18n Resource File	102
EXAMPLE 9-2	Adding a keyword for the new locale	103
EXAMPLE 9-3	Adding a localized string for the locale name	103

Preface

This guide explains how to customize the look and feel of Sun Java™ System Communications Express (hereafter referred to as Communications Express). Although the product architecture permits an almost unlimited customization, this guide focuses on how to perform the most commonly used customizations. In addition, the customizations have been tied together into an application scenario so that examples, code, screen shots, all relate to one another and provide a common frame of reference.

Topics covered in this chapter include

- [“Who Should Read This Book” on page 13](#)
- [“What You Need to Know” on page 13](#)
- [“How this Book is Organized” on page 14](#)
- [“Where to Find Related Information” on page 16](#)
- [“Where to Find this Book Online” on page 16](#)

Who Should Read This Book

You should read this book if you are responsible for administering, configuring, and customizing Sun Java System Communications Express 6.3 at your site. Developers may also find this guide useful for reference.

What You Need to Know

This book assumes a knowledge of Sun Java System Messaging Server software and an understanding of the following:

- JavaScript™
- HTML
- JSP
- Email applications
- Web development
- XML

How this Book is Organized

Organization of the Sun Java System Communications Express 6.3 Customization Guide

TABLE P-1 Organization of the Sun Java System Communications Express 6.3 Customization Guide

Chapter	Description
This Chapter	Describes the audience, requirements, organization, document conventions, and related information.
Chapter 1, “Introduction to Communications Express and Customization”	Provides a high-level overview on how to customize the look and feel of Sun Java System Communications Express 6.3.
Chapter 2, “Customizing General Features in Communications Express”	Provides details about how to customize general features in Sun Java System Communications Express 6.3.
Chapter 3, “Customizing General Features in Calendar”	Provides details about how to customize Calendar in Sun Java System Communications Express 6.3.
Chapter 4, “Customizing General Features in Mail”	Provides details about how to customize Mail in Sun Java System Communications Express 6.3.
Chapter 5, “Customizing User Interface Features in Mail”	Provides details about how to customize User Interface of Mail in Sun Java System Communications Express 6.3.
Chapter 6, “Customizing Advanced Features in Mail”	Provides details about how to perform advanced customization for the mail component. Sun Java System Communications Express 6.3.
Chapter 7, “Customizing General Features in Address Book”	Provides details about how to customize Address Book in Sun Java System Communications Express 6.3.
Chapter 8, “Customizing Domains”	Provides details about customization of specific domains.
Chapter 9, “Localizing Communications Express”	Provides details about how to localize Mail, Calendar and Address Book in Sun Java System Communications Express 6.3.

Monospaced Font

Monospaced font is used for any text that appears on the computer screen or text that you should type. It is also used for file names, distinguished names, functions, and examples.

Bold Monospaced Font

Bold monospaced font is used to represent text within a code example that you should type. For example, you might see something like this:

`./installer`

In this example, `./installer` is what you would type at the command line.

Italicized Font

Italicized font is used to represent text that you enter using information that is unique to your installation (for example, variables). It is used for server paths, names.

For example, throughout this document you will see path references of the form:

msg-svr-base / . . .

The Messaging Server Base (*msg-svr-base*) represents the directory path in which you install the server. The default value of the *msg-svr-base* is `/opt/SUNWmsgsr`.

Italicized font is also used for variables within the synopsis of a command line utility. For example, the synopsis for the `commadmin admin remove` command is:

```
commadmin admin remove -D login -l userid -n domain -w password [-d domain] [-h] [-i
inputfile] [-p port] [-X host] [-s] [-v]
```

In this example, the italicized words are arguments for their associated option. For example, in the `-w password` option, you would substitute the Administrator's password for *password* when you enter the `commadmin admin remove` command.

Default Paths and File Names

The following table describes the default paths and file names used in this book.

TABLE P-2 Default Paths and File Names

Term	Description
<i>msg-svr-base</i>	Represents the base installation directory for Messaging Server. The default value of the <i>msg-svr-base</i> installation is as follows: Solaris™: <code>/opt/SUNWmsgsr</code> Linux: <code>/opt/sun/messaging</code>
<i>cal-svr-base</i>	Represents the base installation directory for Calendar Server. The default value of the <i>cal-svr-base</i> installation is as follows: Solaris: <code>/opt/SUNWics5</code> Linux: <code>/opt/sun/calendar</code>

TABLE P-2 Default Paths and File Names (Continued)

Term	Description
<i>uwc-basedir</i>	Represents the install directory. The directory path is entered for Sun Java System Communications Express 6.3 in the panel “Install Directories” of the Communications Suite installer. Solaris: /opt/SUNWuwc. Linux: /opt/sun/uwc
<i>uwc-deployed-path</i>	Represents the directory where Sun Java System Communications Express 6.3 is deployed. The directory path is entered in the panel “Select Directory to Store Configuration and Data files” of the configurator tool. Solaris: /var/opt/SUNWuwc Linux: /var/opt/sun/uwc
<i>uwc-deployed-path/WEB-INF/config</i>	Represents the directory under which all the Sun Java System Communications Express 6.3 Configuration files are located. Solaris: /var/opt/SUNWuwc/WEB-INF/config Linux: /var/opt/sun/uwc/WEB-INF/config
<i>uwc-deployed-path/WEB-INF/domain</i>	Represents the directory that contains domain specific configuration files. Solaris: /var/opt/SUNWuwc/WEB-INF/domain Linux: /var/opt/sun/uwc/WEB-INF/domain

Where to Find Related Information

In addition to this guide, Sun Java System Communications Express 6.3 comes with supplementary information for administrators as well as documentation for end users and developers. Use the following URL to see all the Communications Express documentation:

<http://docs.sun.com/db/prod>

Where to Find this Book Online

You can view this documentation online in PDF and HTML formats by pointing your browser to the following URL:

<http://docs.sun.com/db/prod/>

Introduction to Communications Express and Customization

Sun Java™ System Communications Express 6.3 provides an integrated web-based communication and collaboration client that caters to the needs of Internet Service Providers, Enterprises, and Original Equipment Manufacturers.

Components of Sun Java System Communications Express 6.3

As a web-based client, Communications Express needs to be deployed on a web container to host and provide access to its various components, and a web browser for presentation. Communications Express consists of three clients - Calendar, Address Book, and Mail. These clients are deployed as a single application on the web container and are collectively referred to as the Unified Web Client.

Calendar uses JSP pages as the presentation layer. These JSP pages can be customized to suit the requirements of the client.

Address Book uses XML/XSL files that contain XSL tags, static HTML and JavaScript. The XSL and JavaScript code are used to display dynamic data. These XSL files can be edited for customizing the Address Book component.

Mail uses HTML, JavaScript, and JSP files. The web client requests mail data from the same proprietary HTTP server and renders them using the JavaScript functions in the downloaded HTML files. All modifications and customization therefore should be done in the HTML and JavaScript files.

Most of the features in Communications Express are fully customizable. Most features can also be customized easily during an upgrade. This guide explains the important features that are customizable.

Customizing General Features in Communications Express

This chapter describes how to customize some of the common elements in Communications Express. It also provides information related to JSP files and their location along with details on the common features that are customizable in Communications Express.

This chapter has the following sections:

- [“Guidelines for Customization” on page 19](#)
- [“Images” on page 22](#)
- [“Text” on page 23](#)
- [“Themes” on page 24](#)
- [“Login Screen” on page 25](#)
- [“Replacing the Logo With a Custom Graphic” on page 25](#)
- [“Banner” on page 27](#)
- [“The Application Bar” on page 27](#)

Guidelines for Customization

Before you start customizing Communications Express, it is important to understand how to edit and deploy the JSP files.

The guidelines for customizing JSP files are as follows:

- Do not edit JSP files directly in your production web server. Instead, make and test changes in a test environment.
- Make a backup copy of the JSP file.
- Edit the JSP file with any HTML and text editor. Typically, HTML editors do not work well for JSP files because of the amount of embedded Java code. In general, when editing a JSP file, be careful not to edit Java code.
- After completing your edits, you can optionally compile the JSP file using a JSP aware IDE. Because the JSP file contains both HTML and Java code, this compilation ensures that you have not introduced any syntax errors.

- Restart the web container for the changes to take effect.

Customization Do's and Don'ts

This section describes various files that are critical and can break your customizations if they are not edited with caution. Some files often contain bug fixes. These changes can be lost if appropriate care is not taken.

Files That Should not be Edited

The following table lists the files that should not be edited. These files often have bug fixes and are not needed for customization in most cases.

TABLE 2-1 List of files that often contain bug fixes

browser.js	compRecipient.js	dtree.js	main.js
form.js	lookup.js	mailui.js	util.js
searchusers.js	setdomain.js	smimeapplet.js	

Files That Should be Edited with Caution

The following table lists files that should be edited with caution. These files are highly readable and self-explanatory. Some of them have HTML code at the bottom. It is recommended that you limit your customization to the HTML portion. Also, do not add or modify any JavaScript functions. If you see a need for a new JavaScript function, create a new file and source it in the HTML code.

TABLE 2-2 List of files that should be edited with caution while customizing Communications Express

applet_fs_lr.html	attach_fs_lr.html	collect_fs_lr.html	comp_fs_lr.html
fldr_fs_lr.jsp	mbox_fs_lr.jsp	msg_fs_lr.jsp	receipt_fs_lr.html
searchmsg_fs_lr.html	setpermission_fs_lr.html	srchresults_fs_lr.html	subscribe_fs_lr.html

Files Related to Global Fonts and Colors

The following table lists files that are related to customizing global styles and fonts. A number of tools are available in the market that can help you find out values of a particular color. It is recommended that you edit these files to achieve the customizations for font and color. However, you can add new styles if you find it impossible to change the styles in these files.

TABLE 2-3 List of files related to global fonts and colors

dtree.css	mail_css_ie5win.css	mail_css_ns6up.css	master-style.css
master-style_ie5up.css	master-style_ns4sol.css	master-style_ns6up.css	master-style_ns4win.css

Customizing Image Files

While customizing images in Communications Express it is recommended that you replace the existing images with the new ones without changing the file names. The file names of the image are read and cached by Communications Express and then rendered. If you want to change the names of the files, you have to change the name in the property files and then restart the web container. Replacing the image files with the new images while maintaining the same file names will avoid restarting of the web container.

List of Customizable and Configurable Files for Calendar and Address Book

The following list of files are customizable and configurable for Calendar and Address Book. It is assumed that the customization is performed before configuring Communications Express. The directories mentioned are relative to `<install root>/lib/config-templates`. If configured, the files are available in the location `/var/opt/SUNWuwc` on Solaris and `/var/opt/sun/uwc` on Linux.

TABLE 2-4 List of Customizable files for Calendar and Address Book

File Location	Customizable Files	Name used in the document
WEB-INF/config	All the configuration files	<code><config-dir></code>
WEB-INF/domain	All <code>i18n.properties</code> files. <code>i18n</code> files are used to internationalize the interface for multiple languages	<code><domain-dir></code>
WEB-INF/skin	<code>themes.properties</code> file	<code><skin-dir></code>
WEB-INF/ui/html/abs	All the <code>.html</code> , <code>.xml</code> and <code>.xsl</code> files	<code><ab-ui-dir></code>
absimx	All files such as <code>.gif</code> , <code>.css</code> , <code>.html</code> files	<code><ab-images-dir></code>
uwc/css	All stylesheets (<code>.css</code> files)	<code><css-dir></code>
uwc/images	All image files (<code>*.gif</code> , <code>*.jpg</code>)	<code><uwc-images-dir></code>
uwc/js	<code>.jsp</code> files	<code><js-dir></code>
uwc/calclient	<code>.jsp</code> files	<code><calclient-dir></code>

TABLE 2-4 List of Customizable files for Calendar and Address Book (Continued)

File Location	Customizable Files	Name used in the document
uwc/common	.jsp files	<common-dir>

Images

All Communications Express images can be customized by specifying the location for these icons in `<skin-dir>/themes.properties`.

To customize the icons, edit the `themes.properties` file to change the names of properties to contain a value that points to the location of the new images. A default value for image location will be used if the corresponding key is not defined in the `themes.properties` file.

Table 2-5 lists the common images that are located in `<uwc-images-dir>`.

TABLE 2-5 Location of the Common images in `<uwc-images-dir>`

Image/Icon	Property Name	Value
Banner Image	<code>uwc-common-bannerImage</code>	<code>Masthead_slUMC.gif</code>
Right Branding Image	<code>uwc-common-RightBrandingImage</code>	<code>logo_sun.gif</code>
Warning Image	<code>uwc-common-WarningImage</code>	<code>Warning_Large.gif</code>
Search Image	<code>uwc-common-SearchImage</code>	<code>LrlSearch_1_wo.gif</code>
Printable Image	<code>uwc-common-PrintableImage</code>	<code>LrlPrintable_1_wo.gif</code>
Import Export Image	<code>uwc-common-ImportExportImage</code>	<code>LrlImpExp_1_wo.gif</code>
Sort down and Non Selected Image	<code>uwc-common-Sort-Down-NonSelected-image</code>	<code>sort_down_nonsel_che.gif</code>
Sort Down and Selected Image	<code>uwc-common-Sort-Down-Selected-image</code>	<code>sort_down_sel.gif</code>
Sort Up and Non Selected Image	<code>uwc-common-Sort-Up-NonSelected-image</code>	<code>sort_up_nonsel.gif</code>
Sort Up and Selected image	<code>uwc-common-Sort-Up-Selected-image</code>	<code>sort_up_sel.gif</code>
Anchor Image	<code>uwc-common-To-Anchor-image</code>	<code>to_anchor.gif</code>
Subscribed Image	<code>uwc-common-Subscribed-image</code>	<code>LrlSub_1.gif</code>
Required Image	<code>uwc-common-Required-image</code>	<code>required.gif</code>
Error Message Icon	<code>uwc-common-Error-Message-icon</code>	<code>Error_Large.gif</code>

TABLE 2-5 Location of the Common images in `<uwc-images-dir>` (Continued)

Image/Icon	Property Name	Value
Warning Message Icon	<code>uwc-common-Warning-Message-icon</code>	<code>Warning_Large.gif</code>
Question Message Icon	<code>uwc-common-Question-Message-icon</code>	<code>Question_Large.gif</code>
Information Message Icon	<code>uwc-common-Info-Message-icon</code>	<code>Info_Large.gif</code>

Text

Communications Express allows you to customize text that is displayed on web pages. This text is both customizable and localizable. You can also configure text on a per domain basis. To customize text, you need to change the `i18n.properties` file that is deployed in `<domain-dir>/domain/en`. The `i18n.properties` file in `<domain-dir>/domain/en` is the default file. If the deployment is configured to work with multiple domains and multiple locales, then you need to create a directory by that domain name under `<domain-dir>` appropriately. For more information on localization, refer to [Chapter 9, “Localizing Communications Express”](#)

▼ To Customize Text for a Domain

To customize the text of a domain called `example.com`, perform the following steps.

- 1 **Create a directory** `<domain-dir>/example.com`
- 2 **Copy** `<domain-dir>/en/i18n.properties` **to** `<domain-dir>/example.com/en`.
This would be the default path of the `i18n.properties` for users in `example.com` domain.
- 3 **Create directories for each supported locale** such as `en_US`, `ja`, `de_DE`, `fr_FR` under `<domain-dir>/example.com/`.
 - a. **Copy** `i18n.properties` **to each domain**.
 - b. **Change the locations accordingly**.

Themes

Communications Express allows you to customize the look and feel of the displayed web pages.

▼ To Create a new Theme

The `<skin-dir>` directory contains subdirectories, which correspond to each skin defined for Communications Express. For example, to create a skin called “christmas” create a “christmas” directory under the `<skin-dir>` directory and perform the following steps:

- 1 **Copy** `themes.properties` **under** `<skin-dir>` **directory to** `<skin-dir>/christmas`. `themes.properties` **file that defines the various parameters of this theme.**
- 2 **Change the locations in** `themes.properties` **to point to the customized style sheets and images.**
- 3 **For help on various parameters in this file, refer to Appendix D of the Communications Express Administration Guide. Look for “**`themes.properties`**” under the “Configuration Parameters Reference” section.**
- 4 **To configure the** `christmas` **theme, go to** `<domain-dir>` **and add the parameter** `uwc-user-attr-SunUCTheme-<locale-name>=christmas` **in the** `uwcdomainconfig.properties` **file. Here, <locale-name> is the name of the locale on which you want to set the theme .**
- 5 **Restart the web server to apply the changes.**

▼ To Modify a Skin

The `<skin-dir>` directory contains subdirectories, which correspond to each skin defined for Communications Express. You can also modify the properties associated with a particular skin by editing the `themes.properties` file for the skin.

- 1 **Go to the skin directory that you want to update. For example, to update the skin called “christmas” change to** `<skin-dir>/christmas` **directory.**
- 2 **The** `<skin-dir>/christmas` **contains a file called** `themes.properties`. **This file defines the various parameters of the skin.**
- 3 **Update the required parameters in the** `themes.properties` **file.**

Note – For help on various parameters in this file, refer to Appendix D of the Communications Express Administration Guide. Look for “themes.properties” under the “Configuration Parameters Reference” section.

4 Restart the web server to apply changes.

Login Screen

This section describes how to modify the Communications Express Login screen shown in [Figure 2-1](#).

You can replace the logo with a custom graphic on the Communications Express Login screen.

To modify the Communications Express Login Screen, you need to edit `<skin-dir>/themes.properties` file. The relevant .jsp file is `login.jsp`.



FIGURE 2-1 Communications Express Login Screen.

Replacing the Logo With a Custom Graphic

To replace the default Communications Express logo with a custom logo:

- Replace the logo with a custom graphic
- Change the destination of link



FIGURE 2-2 Sun Logo and Link

Example: Logo Modification

The example shown in [Figure 2-3](#) replaces the Sun logo with a custom logo having different dimensions.



FIGURE 2-3 Example Of a Corner Logo



Copyright © 2002 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers. Sun, Sun Microsystems, the Sun logo, Java and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Federal Acquisitions: Commercial Software - Government Users Subject to Standard License Terms and Conditions.

FIGURE 2-4 Login Screen with Customized Logo

Changing the Title Text



FIGURE 2-5 Communications Express Title Text

To change the title text for Communications Express:

- Replace a title graphic with custom graphic
- Change the title text

Banner

You can modify the left image and the right image in the banner. Figure 2–6, Figure 2–7 and Figure 2–8, Figure 2–9 display the banner before and after customization.



FIGURE 2–6 Left Banner Image Before Customization



FIGURE 2–7 Customized Left Banner Image



FIGURE 2–8 Right Banner Image Before Customization



FIGURE 2–9 Customized Right Banner Image

The Application Bar

You can modify the application bar by editing the `Ca\ApplBarNormal.jsp` file located at `<calclient-dir>/Ca\ApplBarNormal.jsp`

By default, the order in which the application tab appears, provided all services are enabled, is 'Mail', 'Calendar', 'Address Book' and 'Options'. This order can be changed to suit your requirements.



FIGURE 2-10 Default application tab order

[Example 2-1](#) shows the default code that displays the default application tab order.

EXAMPLE 2-1 Code for Customizing the Application Tab Order

```
<td class="Tab1Gutter">
  <table border="0" cellspacing="0" cellpadding="0" class="Tab1Tbl">
    <tr>
      <td colspan="12">
        
      </td>
    </tr>
    <tr>
      <jato:content name="MailTab">
        <td class="Tab1Gutter" colspan="3">
          
        </td>
      </jato:content>
      <td rowspan="3"
        class="Tab1Sel">
        <div class="Tab1Sel">
          <span class="Tab1LblSel"
            title="<%= getLocalizedLabel(session,
              "uwc-calclient-toolbar-tooltip-Calendar",
              "Calendar")
            %>">
            <%= getLocalizedLabel(session,
              "uwc-calclient-toolbar-Calendar",
              "Calendar")
            %>
          </span>
        </div>
      </td>
      <td rowspan="3"
        class="Tab1Sel">
        
      </td>
```

EXAMPLE 2-1 Code for Customizing the Application Tab Order (Continued)

```

</td>
<td class="Tab1Gutter"
    colspan="7">
    
</td>
</tr>
<tr>
    <jato:content name="MailTab">
    <td class="Tab1NotSel">
        <div class="Tab1NotSel">
            <a href="<%= getContextURI(request) %>/mailclient/Mail"
                title="<%= getLocalizedLabel(session,
                    "uwc-calclient-toolbar-tooltip-Mail",
                    "Mail")
                    %>"
                class="Tab1LblNormal"
                onmouseover="status='Mail';
                return true;
                "onmouseout="status='';">
                <%= getLocalizedLabel(session,
                    "uwc-calclient-toolbar-Mail",
                    "Mail")
                    %>
            </a>
        </div>
    </td>
    <td class="Tab1NotSel">
        
    </td>
    <td class="Tab1Gutter">
        
    </td>
    </jato:content>
    <td class="Tab1Gutter">
        </td>
<td class="Tab1NotSel" nowrap>
  <div class="Tab1NotSel">
    <a href="<%= getContextURI(request)%> /abclient/AddressBook"
      title="<%= getLocalizedLabel(session,
                                "uwc-calclient-toolbar-tooltip-AddressBook",
                                "Address Book")
      %>"
      class="Tab1LblNormal"
      onmouseover="status='Address Book';
      return true;"
      onmouseout="status='';">
    <%= getLocalizedLabel(session,
                          "uwc-calclient-toolbar-AddressBook",
                          "Address Book")
    %>
  </a></div>
</td>
<td class="Tab1NotSel">
  </td>
<td class="Tab1Gutter">
  
</td>
<td class="Tab1NotSel">
  <div class="Tab1NotSel">
    <a href="<%= getContextURI(request) %> /base/CalendarPreferences"
      title="<%= getLocalizedLabel(session,
                                "uwc-calclient-toolbar-tooltip-Options",
                                "Options")
      %>"
      class="Tab1LblNormal"
      onmouseover="status='Options';
      return true;"
      onmouseout="status='';">
    <%= getLocalizedLabel(session,
                          "uwc-calclient-toolbar-Options",
                          "Options")
    %>
  </a>
</div>
```

EXAMPLE 2-1 Code for Customizing the Application Tab Order (Continued)

```

        </td>
        <td class="Tab1NotSel">
            
        </td>
        <td class="Tab1Gutter">
            
        </td>
    </tr>
<tr>
    <td colspan="3"
        class="Tab1Gutter">
        
    </td>
    <td colspan="7"
        class="Tab1Gutter">
        
    </td>
</tr>
</table>
</td>

```

The application tab order can be customized to display in the order, Address Book, Mail, Calendar and Options.



FIGURE 2-11 Changing the Application Tab Order to Display Address Book, Mail, Calendar and Options

To change the order of Application tabs for example 'Address Book', 'Mail', 'Calendar', 'Options' rearrange the code shown in [Example 2-2](#) to the following:

EXAMPLE 2-2 Customizing the order of the application tabs

```
<td class="Tab1Gutter">
  <table border="0" cellspacing="0" cellpadding="0" class="Tab1Tbl">
    <tr>
      <td colspan="12">
        </td></tr><tr> <td class="Tab1Gutter">
          </td>
        <td class="Tab1NotSel" nowrap> <div class="Tab1NotSel">
          <a href="<%= getContextURI(request) %>/abclient/AddressBook"
            title="<%= getLocalizedLabel(session,
              "uwc-calclient-toolbar-tooltip-AddressBook", Address Book) %>"
            class="Tab1LblNormal"
            onmouseover="status='Address Book'; return true;"
            onmouseout="status='';">
            <%= getLocalizedLabel(session,
              "uwc-calclient-toolbar-AddressBook",
              "Address Book") %></a></div>
        </td>
      </td>
    </tr>
  </table>
</td>

<jato:content name="MailTab">
  <td class="Tab1NotSel">
    <div class="Tab1NotSel">
      <a href="<%= getContextURI(request) %>/mailclient/Mail"
        title="<%= getLocalizedLabel(session,
          "uwc-calclient-toolbar-tooltip-Mail",
          "Mail") %>"
        class="Tab1LblNormal"
        onmouseover="status='Mail'; return true;"
        onmouseout="status='';">
        <%=getLocalizedLabel(session, "uwc-calclient-toolbar-Mail", "Mail") %>
      </a></div>
    </td><td class="Tab1NotSel"></td>
    <td class="Tab1Gutter"></td>
  </td>
</jato:content>
<jato:content name="MailTab">
  <td class="Tab1Gutter" colspan="3">
    </td>
  </jato:content> <td rowspan="3" class="Tab1Sel">
    <div class="Tab1Sel"><span class="Tab1LblSel"
      title="<%= getLocalizedLabel(session,
        "uwc-calclient-toolbar-tooltip-Calendar", "Calendar") %>"
        <%= getLocalizedLabel(session, "uwc-calclient-toolbar-Calendar",
          "Calendar") %>
    </span></div>
  </td>
</div>
```

EXAMPLE 2-2 Customizing the order of the application tabs *(Continued)*

```

</td><td rowspan="3" class="Tab1Sel">
    </td>
    <td class="Tab1Gutter" colspan="7">
</td>
</tr> <tr>
<td class="Tab1NotSel">
    </td>
    <td class="Tab1Gutter">
        </td>
    <td class="Tab1NotSel">
        <div class="Tab1NotSel">
            <a href="<%= getContextURI(request) %>/base/CalendarPreferences"
                title="<%= getLocalizedLabel(session,
                    "uwc-calclient-toolbar-tooltip-Options", "Options") %>"
                class="Tab1LblNormal" onmouseover="status='Options';
                return true;" onmouseout="status='';">
                <%= getLocalizedLabel(session, "uwc-calclient-toolbar-Options",
                    "Options") %></a></div>
        </td> <td class="Tab1NotSel">
            </td>
            <td class="Tab1Gutter">
                </td>
        </tr>
        <tr>
        <td colspan="3" class="Tab1Gutter">
            </td>
            <td colspan="7" class="Tab1Gutter">
                </td>
        </tr>
</table></td>

```


Customizing General Features in Calendar

The look and feel of the calendar is provided by a set of style sheets, images and localizable text strings. These strings pertain to the text that is displayed on the pages. The code examples show how to modify icons, text and stylesheets for Calendar.

Note – This chapter does not provide code samples for all customization scenarios in Calendar. The code samples have been provided only for illustrative purposes. This cannot be used directly for customization.

These are some common customization scenarios:

- “Icons” on page 35
- “Views” on page 45
- “Pop ups” on page 50
- “Full Page Error” on page 51

Icons

To customize icons and images in Calendar, you need to edit various JSP files and property files. The following sections describe various customizations that can be done in Calendar.

- “The Order of Calendar Toolbar Icons” on page 36
- “Icons in Various Calendar Views” on page 41
- “The Toolbar Images” on page 42
- “Events/Task Related Informative Icons” on page 43
- “The Error Icon for Full Page Errors” on page 43
- “Icons in Pop ups” on page 44
- “Search for Events” on page 44

The Order of Calendar Toolbar Icons

To change the order of the calendar toolbar icons, you need to modify the `<calclient-dir>/CalToolBarNormal.jsp` file.

Figure 3-1 shows the default order of the Calendar Toolbar Icons.



FIGURE 3-1 Order of Calendar Toolbar Icons

Example 3-1 shows the default code used to display the Calendar Toolbar Icons in the order, New Event, New Task, Check Availability, Search for Calendar, Printable and Import/Export. The `CalToolBarNormal.jsp` file contains specific sections of code to display and perform actions each of the components in the tool bar. To change the order of the toolbar icons, you need to change the order in which they appear in the JSP file.

EXAMPLE 3-1 Default Code that Displays the Calendar Toolbar Icons

The code below is common to the `CalToolBarNormal.jsp` file.

```
<div class="Toolbar">
a name="toolbar">
</a>
    <script language="JavaScript" src="../uwc/js/define_images_cal.js"></script>
<table border="0" cellspacing="0" cellpadding="0">
    <tr>
```

The code for New Event icon starts here:

New Event

```
<td nowrap>
<a href="javascript:void(0)"
    accesskey="e"
    onMouseOver="over('newEvent')"
    onMouseOut="out('newEvent')"
    onClick="openWinAutoHeight
        ('<jato:text name="NewEventUrl" escape="false"/>', 'eventWin', 'scrollbars=yes,
            resizable=yes,width=700'); return false;">
"
    width="24"
    height="24"
    align="absmiddle"
```

EXAMPLE 3-1 Default Code that Displays the Calendar Toolbar Icons *(Continued)*

```

border="0"
title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-NewEvent",
    "New Event") %>"
alt="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-NewEvent",
    "New Event") %>"
</a>
<a href="javascript:void(0)"
    title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-NewEvent",
        "New Event") %>"
    class="ToolLbl"
    onClick="openWinAutoHeight('<jato:text name="NewEventUrl" scape="false"/>',
        'eventWin', 'scrollbars=yes,resizable=yes,width=700'); return false;">
    <%= getLocalizedLabel(session, "uwc-calclient-toolbar-NewEvent", "New Event") %>
</a>
</td>

```

The code for New Task Event starts here:

```

New Task
<td class="ToolbarItem" nowrap>
<a href="javascript:void(0)"
    accesskey="t"
    onMouseOver="over('newTask')"
    onMouseOut="out('newTask')"
    onClick="openWinAutoHeight
        ('<jato:text name="NewTaskUrl" escape="false"/>', 'task', 'scrollbars=yes,
            resizable=yes,width=700'); return false;">
    "
        width="24"
        height="24"
        align="absmiddle" border="0"
        title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-NewTask",
            "New Task") %>"
        alt="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-NewTask",
            "New Task") %>"
    </a>
    <a href="javascript:void(0)"
        title="<%= getLocalizedLabel(session "uwc-calclient-toolbar-tooltip-NewTask",
            "New Task") %>"
        class="ToolLbl"
        onClick="openWinAutoHeight('<jato:text name="NewTaskUrl" escape="false"/>',
            'task', 'scrollbars=yes,resizable=yes,width=700'); return false;">
        <%= getLocalizedLabel(session, "uwc-calclient-toolbar-NewTask", "New Task") %>
    </a>
</td>

```

EXAMPLE 3-1 Default Code that Displays the Calendar Toolbar Icons (Continued)

The code for Check Availability icon starts here:

```

Check Availability
<td class="ToolbarItem" nowrap>
<a href="javascript: void(0)"
  accesskey="a" onMouseOver="over('checkAvail')"
  onMouseOut="out('checkAvail')"
  onClick="openWinAutoHeight
    ('<jato:text name="AvailabilityUrl" escape="false"/>'availability','scrollbars=yes,
      resizable=yes,width=800');return false;">

  "
    width="24"
    height="24"
    align="absmiddle"
    border="0"
    title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-CheckAvailability",
      "Check Availability") %>"
    alt="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-CheckAvailability",
      "Check Availability") %>"
  </a>

  <a href="javascript: void(0)"
    title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-CheckAvailability",
      "Check Availability") %>"
    class="ToolLbl"
    onClick="openWinAutoHeight('<jato:text name="AvailabilityUrl" escape="false"/>'
      , 'availability','scrollbars=yes, resizable=yes,width=800'); return false;">
      <%= getLocalizedLabel(session, "uwc-calclient-toolbar-CheckAvailability",
        "Check Availability") %>
    </a>
</td>

```

The code for Search For Calendar icon starts here:

```

Search for Calendar
<!-- Show SearchCalendar if and only if the parent view is not manage calendars -->
<jato:content name="ShowSearchCalendars">
  <td class="ToolbarItem" nowrap>
    <a href="javascript: void(0)" accesskey="s"
      onMouseOver="over('search')"
      onMouseOut="out('search')"
      onClick="openSearchForCalendarsPopup(); return false;">
      "
      </a>
    </td>
  </jato:content>

```

EXAMPLE 3-1 Default Code that Displays the Calendar Toolbar Icons (Continued)

```

        width="24"
        height="24"
        align="absmiddle"
        border="0"
        title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-SearchCalendar",
            "Search for Calendar") %>"
        alt="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-SearchCalendar",
            "Search for Calendar") %>"
    </a>
    <a href="javascript: void(0)"
        title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-SearchCalendar",
            "Search for Calendar") %>"
        class="ToolLbl"
        onClick="openSearchForCalendarsPopup();
        return false;">
        <%= getLocalizedLabel(session, "uwc-calclient-toolbar-SearchCalendar",
            "Search for Calendar") %>
    </a>
</td>
</jato:content>

```

The code for Printable icon starts here:

Printable

```

<!-- Show Printable f and only if the parent view is not invitations -->
<jato:content name="ShowPrintable">
    <td class="ToolbarItem" nowrap>
        <a href="javascript: void(0)"
            onClick="openWinAutoHeight('<jato:text name="PrintUrl" escape="false"/>', 'print',
            'menubar=yes,
            scrollbars=yes,
            resizable=yes,
            width=700,
            height=650');
            return false;"
            accesskey="p"
            class="ToolLbl" onMouseOver="over('printable')"
            onMouseOut="out('printable')">
            "
                width="24"
                height="24"
                align="absmiddle"
                border="0"
                title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-Printable",

```

EXAMPLE 3-1 Default Code that Displays the Calendar Toolbar Icons (Continued)

```

        "Display a Printable Page") %>"
    alt="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-Printable",
        "Display a Printable Page") %>"></a>
<a href="javascript:void(0)"
    onClick="openWinAutoHeight('<jato:text name="PrintUrl" escape="false"/>',
        'print',
        'menubar=yes,
        scrollbars=yes,
        resizable=yes,
        width=700,
        height=650');
    return false;"
    title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-Printable",
        "Display a Printable Page") %>"
    class="ToolLbl"><%= getLocalizedLabel(session, "uwc-calclient-toolbar-Printable",
        "Printable") %></a></td>
</jato:content>

```

The code for Import/Export icon starts here:

Import/Export

```

    <td class="ToolbarItem" nowrap>
<a href="javascript: void(0)"
    accesskey="i"
    onMouseOver="over('importExport')"
    onMouseOut="out('importExport')"
    onClick="openWinAutoHeight('<jato:text name="ImportExportUrl" escape="false"/>', 'importexport',
    scrollbars=yes,
    resizable=yes,width=700'); return false;"
"
    width="24"
    height="24"
    align="absmiddle"
    border="0"
    title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-ImportExport",
        "Import and Export Calendar") %>"
    alt="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-ImportExport",
        "Import and Export Calendar") %>"
</a>
<a href="javascript: void(0)"
    title="<%= getLocalizedLabel(session, "uwc-calclient-toolbar-tooltip-ImportExport",
        "Import and Export Calendar") %>"
    class="ToolLbl"
    onClick="openWinAutoHeight('<jato:text name="ImportExportUrl" escape="false"/>',

```

EXAMPLE 3-1 Default Code that Displays the Calendar Toolbar Icons (Continued)

```

'importexport','scrollbars=yes,resizable=yes,width=700'); return false;">
<%= getLocalizedLabel(session, "uwc-calclient-toolbar-ImportExport",
                                "Import/Export") %>
</a>

```

These instructions mark the end of the code.

```

</td>
</tr>
</table>
</div>

```

▼ To Change the Order of the Icons in the Calendar Toolbar

In order to change the order of the icons, you need to move the sections specific to each icons around in the order you want them to appear. Please refer to [Example 3-1](#), to change the order of the icons. For example to change the order of the icons from:

 New Event  New Task  Check Availability  Search for Calendar  Printable  Import/Export

to:

 New Task  New Event  Printable  Check Availability  Import/Export  Search for Calendar

perform the following steps:

- 1 Cut the section of code for New Task event and paste it above the instructions for New Event.
- 2 Cut the section of code for Printable and paste it after the instructions for New Event.
- 3 Cut the section of code for Search for Calendar and paste it after the instructions for Import/Export.

Icons in Various Calendar Views

To customize the icons in Calendar views, you need to modify the `themes.properties` file located at `<skin-dir>`.

You can customize toolbars for all the main calendar views such as the following:

- Day view
- Week view
- Month view

- Year view
- Events view
- Invitations view
- Tasks view
- Manage calendar view

Table 3–1 shows the list of customizable icons: .

TABLE 3–1 Customizing Icons

Image	Property name	Description	Value
Forward Icon	<i>uwc-calclient-Forward-Arrow-image</i>	Icon to view events or tasks in the next day, week, month, or year in a calendar	<code>forward.gif</code>
Backward Icon	<i>uwc-calclient-Back-Arrow-image</i>	Icon to view events or task in the previous day, week, month, or year in a calendar	<code>back.gif</code>
Select all Icon	<i>uwc-calclient-Select-All-image</i>	Icon to select all events or tasks on a page	<code>check_all.gif</code>
Unselect all Icon	<i>uwc-calclient-Deselect-All-image</i>	Icon to unselect all events or tasks on a page	<code>uncheck_all.gif</code>
Sort Down Non Selected Image	<i>uwc-common-Sort-Down-NonSelected-image</i>	Icon to sort down, in descending order, unselected events or tasks on a page	<code>sort_down_nonsel_che.gif</code>
Sort Down Selected Image	<i>uwc-common-Sort-Down-Selected-image</i>	Icon to sort down, in descending order, selected events or tasks on a page	<code>sort_down_sel.gif</code>
Sort Up Non Selected Image	<i>uwc-common-Sort-Up-NonSelected-image</i>	Icon to sort up, in ascending order, selected events or tasks on a page	<code>sort_up_nonsel.gif</code>
Sort Up Selected Image	<i>uwc-common-Sort-Up-Selected-image</i>	Icon to sort up, in ascending order, selected events or tasks on a page	<code>sort_up_sel.gif</code>

The Toolbar Images

The property key names that define the location of the toolbar images are mentioned in Table 3–2

TABLE 3–2 Customizing the Calendar Toolbar Icons

Icon	Property Name	Description	Value
New Event icon	<i>uwc-calclient-NewEventImage</i>	Icon to create a new event	<code>LrlNewEvent_1_wo.gif,</code>

TABLE 3–2 Customizing the Calendar Toolbar Icons (Continued)

Icon	Property Name	Description	Value
New Task Icon	<i>uw-calc-client-NewTaskImage</i>	Icon to create a new task	LrlNewTask_1_wo.gif
Check Availability Icon	<i>uw-calc-client-CheckAvailabilityImage</i>	Icon to check availability of individuals before scheduling an event with them	LrlCheckAvail_1_wo.gif
Search Calendar Icon	<i>uw-calc-client-SearchImage</i>	Icon to search for a calendar	LrlSearch_1_wo.gif
Printable Icon	<i>uw-calc-client-PrintableImage</i>	Icon to print calendars, events, or tasks	LrlPrintable_1_wo.gif
Import/Export Icon	<i>uw-calc-client-ImportExportImage</i>	Icon to import and export calendar events and tasks between different calendars	LrlImpExp_1_wo.gif

Events/Task Related Informative Icons

The property key names that define the location of the task related informative icons mentioned in [Table 3–3](#)

You can customize informative icons or images such as the ones used for recurring, reminding, and public tasks for tasks and events.

TABLE 3–3 Customizing Task Related Informative Icons

Icon	Property Name	Description	Value
Notify Image	<i>uw-calc-client-calc-client-NotifyImage</i>	Icon that specifies that the creator of the event and the attendees will receive a notification before the event. The notification period can be specified by the creator of the event.	LrlNotify_1.gif
Recurring Image	<i>uw-calc-client-RecurringImage</i>	Icon that specifies that this is a recurring event	LrlRecur_1.gif
Public Image	<i>uw-calc-client-PublicImage</i>	Icon that specifies that this event is public and can be viewed by anyone visiting this calendar.	LrlPrvPub_1.gif

The Error Icon for Full Page Errors

“The Error Icon for Full Page Errors” on page 43 shows the customizable error icons for Full Page Errors

TABLE 3-4 Customizing Error Icon for Full Page Errors

Icon	Property Name	Description	Value
Error Icon	<i>uw-calcclient-Error-Message-icon</i>	Icon that specifies that an error has occurred.	Error_Large.gif

Icons in Pop ups

Table 3-5 shows the customizable icons located in uw-calcclient/SearchCalendarData.jsp.

TABLE 3-5 Customizing Icons in Pop ups

Icon	Property Name	Description	Value
Select All	<i>uw-calcclient-Select-All-image</i>	Icon to select all events or tasks on a page	check_all.gif
Unselect All	<i>uw-calcclient-Deselect-All-image</i>	Icon to unselect all events or tasks on a page	uncheck_all.gif
Subscribed Icon	<i>uw-calcclient-Subscribed-image</i>	Icon to subscribe to a specified calendar	LrlSub_1.gif

Search for Events

The file name including path is <calcclient-dir>/CalSearchBarPagelet.jsp. Communications Express allows you to customize the Search for events label as shown in [“Search for Events” on page 44](#)



This is applicable for the following views:

- Day view
- Week view
- Month view
- Year view
- Events manager

Besides these, you can change the text that appears besides the search events text box and also customize the overall layout.

Text that Appears Besides the Search Events Text Box

To change the introductory text that is displayed adjacent to the search events text box, change the property `uwc-calclient-toolbar-SearchForEvents` in the resource bundle of `<domain-dir>/en/i18n.properties`.

[Example 3–2](#) shows the code after modifying the resource bundle.

EXAMPLE 3–2 Code After Modifying the Resource Bundle

```
uwc-calclient-toolbar-SearchForEvents=<Customized Text>
```

Views

These examples show how to modify views:

- [“Tasks Search View” on page 45](#)
- [“The Task Column Order in Tasks Printable View” on page 46](#)
- [“Tasks Error View” on page 49](#)
- [“Events view” on page 49](#)
- [“Tasks View” on page 50](#)

Tasks Search View

- The file name including path that pertains to tasks search view is `<calclient-dir>/tasksSearch.jsp`
- The common components present in `tasksSearch.jsp` are: Banner, Application Bar, and Calendar Tool Bar
- In the `tasksSearch.jsp` you can customize:
 - The overall layout and look and feel
 - Icons or images such as `select_all.gif`, `unselect_all.gif`, `sort_up.gif`, `sort_down.gif`, `sort_selected_up.gif`, `sort_selected_down.gif`
 - Task related informative icons or images such as `recurring.gif`, `reminding.gif`, `public task.gifs`
 - The order of the table column
 - The edit task and view task popup window sizes and styles

The Task Column Order in Tasks Printable View

The file pertaining to customizing the column order in tasks is `<calclient-dir>/tasksPrint.jsp`

The common components present are Banner and Application bar.

You can customize the table column ordering.

The `tasksPrint.jsp` is located at `<calclient-dir>/tasksPrint.jsp`

Example 3–3 shows the code used to customize Tasks print view in a table with the columns in the order - 'Priority', 'Title', 'Due Date', 'Status', 'Type'

EXAMPLE 3–3 Default Code that Displays the Table Column Order

Code to display the header of the table containing the tasks created by a user.

The header row for the table.

```
<tr>
```

Code for the displaying the Priority header starts here.

```
<th class="TblThCl1" scope="col" nowrap>
  <strong>
    <%= getLocalizedLabel(session, "uwc-calclient-tasks-Priority", "Priority") %>
  </strong>
</th>
```

Code for displaying the Title header starts here.

```
<th width="35%" scope="col" nowrap>
  <strong>
    <%= getLocalizedLabel(session, "uwc-calclient-tasks-Title", "Title") %>
  </strong>
</th>
```

Code for displaying the Due Date header starts here.

```
<th scope="col" nowrap>
  <strong>
    <%= getLocalizedLabel(session, "uwc-calclient-tasks-DueDate", "Due Date") %>
  </strong>
</th>
```

Code for displaying the Status header starts here.

EXAMPLE 3-3 Default Code that Displays the Table Column Order (Continued)

```

<th scope="col" nowrap>
  <strong>
    <%= getLocalizedLabel(session, "uwc-calclient-tasks-Status", "Status") %>
  </strong>
</th>

```

Code for displaying the Type header starts here.

```

<th scope="col" nowrap>
  <strong>
    <%= getLocalizedLabel(session, "uwc-calclient-tasks-Type", "Type") %>
  </strong>
</th>

```

Code to close the table row for the header columns.

```

</tr>

```

To change the order in which the columns appear, rearrange the above code for each column in the order of your choice.

Code to check if there are any tasks created by the user. If there are no tasks to display, the following message is displayed.

"This view does not have any tasks to display. Choose another filter or search for tasks".

```

<jato:content name="ZeroTasks">
<tr>
  <td class="TblTdCl1Lst" colspan="6">
    <%= getLocalizedLabel(session, "uwc-calclient-tasks-NoTasksToDisplay-message",
      "This view does not have any tasks to display.
      Choose another filter or search for tasks.") %>
  </td>
</tr>
</jato:content>

```

These code instructions are executed if there are one or more tasks. The display of the tasks is rendered by the `jato:tilableView` tag. This tag takes care of iterating through the users tasks and displaying them.

```

<jato:content name="NonZeroTasks">
  <jato:tilableView name="TasksTiledView"
    type="com.sun.uwc.calclient.TasksTiledView">
    <jato:hidden name="TaskCalID"/>
    <jato:hidden name="TaskUID"/>
    <jato:hidden name="TaskRID"/>
  </jato:tilableView>
</jato:content>

```

EXAMPLE 3-3 Default Code that Displays the Table Column Order (Continued)

Code to display the Priority of the task.

```
<td <jato:content name="FirstColumnStyle"/> align="center">
<jato:text name="Priority" escape="false"/>
</td>
```

Code to display the Title of the task. Along with the title, this code also displays other properties of the task such as

- If the task has a notification
- If the task is recurring
- If the task is a public event
- Alarms associated with a Task if any
- Is this a public task.

```
<jato:content name="IsConfidential">
  <td <jato:content name="TitleColumnStyle"/>
    <%= getLocalizedLabel(session, "i18nModel", "uwc-calclient-view-Busy", "Busy") %></td>
</jato:content>
<jato:content name="IsNotConfidential">
  <td <jato:content name="NormalColumnStyle"/>>
<strong>
  <jato:text name="Title"/>
</strong><br>
<jato:content name="HasAlarm">
  "
    width="12"
    height="12"
    border="0"
    hspace="2"
    align="absmiddle"
    alt="<%= getLocalizedLabel(session, "uwc-calclient-tasks-Notify", "Notify") %>"
    title="<%= getLocalizedLabel(session, "uwc-calclient-tasks-Notify", "Notify") %>"
  >
</jato:content>
<jato:content name="IsRecurring">
"
  width="12" height="12" border="0" align="absmiddle" hspace="2"
  title="<%= getLocalizedLabel(session, "uwc-calclient-tasks-Recurring", "Recurring") %>"
  alt="<%= getLocalizedLabel(session, "uwc-calclient-tasks-Recurring", "Recurring") %>"
  >
</jato:content>
<jato:content name="IsPublic">
  "
    width="12"
```

EXAMPLE 3-3 Default Code that Displays the Table Column Order (Continued)

```

        height="12"
        border="0"
        hspace="2"
        title="<%= getLocalizedLabel(session, "uwc-calclient-tasks-Public","Public") %>"
        alt="<%= getLocalizedLabel(session, "uwc-calclient-tasks-Public","Public") %>"
    </jato:content>
    <jato:content name="IsExternal"><br>
    [<jato:text name="OwnerName"/>&nbsp;<jato:text name="CalendarName"/>]
    </jato:content>
</td>

```

Code to display the Due Date.

```

</jato:content>
    <td <jato:content name="NormalColumnStyle"/>>
        <jato:text name="DueDate"/>&nbsp;<jato:text name="DueTime"/>
    </td>

```

Code to display the Status of the task.

```

<td <jato:content name="NormalColumnStyle"/>><jato:text name="PercentStatusAsText" escape="false"/></td>
<td <jato:content name="NormalColumnStyle"/>><jato:text name="Category" escape="false"/></td>

```

The tiledView tag marks the end of the display.

```

    </tr>
    </jato:tiledView>
</jato:content>

```

In order to change the order in which the Tasks appear in the task printable view, you need to move the sections specific to each of the components in the order you want them to appear.

Tasks Error View

To modify the tasks error view, you need to edit the file `<calclient-dir>/tasksError.jsp`.

Events view

- To modify the events view, edit the following JSP file.
`<calclient-dir>/EventManager.jsp`
- To modify the search events view, edit the following JSP file.

<calclient-dir> /eventsSearch.jsp

- To modify the anonymous events view, edit the following JSP file.
<calclient-dir> /EventManagerAnon.jsp
- To modify the anonymous search events view, edit the following JSP file.
<calclient-dir> /eventsSearchAnon.jsp
- To modify the anonymous Error view, edit the following JSP file.
<calclient-dir> /eventsanonError.jsp

To modify the printable images view, edit the following property in the themes.properties file under the skin directory. You can change the image and then replace the changed image in the root folder.

uwc-common-PrintableImage=../uwc/images/LrlPrintable_1_wo.gif.

Tasks View

To modify the tasks view, you need to modify the file <calclient-dir> /tasks.jsp In a task view, you can modify:

- Layout
- Order in which the various sections appear
- Alignment of the items and images

Pop ups

Table 3–6 shows the pop up names and the relevant JSP files to be modified for customization.

TABLE 3–6 Customizing Pop ups

Pop up Name	jsp files to be modified
New and Edit Event Pop up	<calclient-dir> /EditEvent.jsp
	<calclient-dir> /EditEventData.jsp
	<js-dir> /DeleteEventConfirmRepeatSelection.jsp
	<js-dir> /SaveEventConfirmRepeatSelection.jsp
	<js-dir> /EditEventButtons.jsp
	<js-dir> /NewEventButtons.jsp

TABLE 3-6 Customizing Pop ups (Continued)

Pop up Name	jsp files to be modified
View Calendar Pop up	<code><calclient-dir>/ViewCalendar.jsp</code> <code><calclient-dir>/ViewCalendarData.jsp</code> <code><js-dir>/ViewCalendarButtons.jsp</code>
New and Edit Calendar Group Pop up	<code><calclient-dir>/EditCalGroup.jsp</code> <code><calclient-dir>/EditCalGroupData.jsp</code> <code><js-dir>/EditCalGroupButtons.jsp</code> <code><js-dir>/NewCalGroupButtons.jsp</code>
Subscribe to Calendars Pop up and Search for Calendars Pop up	<code><calclient-dir>/SearchCalendar.jsp</code> <code><calclient-dir>/SearchCalendarData.jsp</code> <code><calclient-dir>/SearchCalendarInitial.jsp</code> <code><js-dir>/SearchCalendarButtons.jsp</code> <code><js-dir>/SearchNViewCalendarButtons.jsp</code>

Full Page Error

To modify the Error page, you need to edit the file `<common-dir>/FullPageError.jsp`

You can customize the error page for calendar views such as:

- Day view
- Week view
- Month view
- Year view
- Events manager
- Invitations view
- Tasks view
- Manager calendar view

Customizing the Layout for Full Page Errors

Full page errors appear for a variety of reasons. Here are a few examples:

- Misconfiguration
- Network Failure
- Failure in connecting to the LDAP server

This page displays the following information:

- Full Page Error Summary
- Full Page Error Sub-Title
- Full Page Error Description
 - `FullPageErrorSummary` refers to Full Page Error Summary
 - `FullPageErrorSubTitle` refers to Full Page Error Sub-Title
 - `FullPageErrorDescription` refers to Full Page Error Description

These keys are replaced with the actual information provided by the server depending on the type or cause of error. To customize the information that is displayed for each of these keys:

Edit the `i18n.properties` file, look for the following property names and change the value of the property as follows:

`<per-view-key>-FullPageError-<ReasonForError>-summary=<Summary to be displayed>`

`<per-view-key>-FullPageError-<ReasonForError>-subtitle=<Sub Title to be displayed>`

`<per-view-key>-FullPageError-<ReasonForError>-description=<Description to be displayed>`

For example,

`calclient-error-dayviewview-FullPageError-
CalendarNotAvailable-summary=Calendar Not Available`

`calclient-error-dayviewview-FullPageError-
CalendarNotAvailable-subtitle=Could Not Display View`

`calclient-error-dayview-FullPageError-
CalendarNotAvailable-description=The selected calendar was either deleted or does
not exist or you do not have permissions to view it. Select another calendar.`

Customizing General Features in Mail

This chapter includes information on the customizable features in Mail. This chapter has the following sections:

- “Customization Scenarios” on page 53
- “Understanding the File Layout in Mail” on page 54
- “Understanding Basic Interfaces in Mail” on page 55

Customization Scenarios

Before you customize Mail, you need to decide your customization goals. There are several common customization scenarios.

- “Customizing the Skin to Modify Global Look and Feel” on page 53
- “Customizing Text” on page 53
- “Customizing Mail by Domain” on page 54
- “Customizing Mail Icons” on page 54
- “Adding Extra Features” on page 54

Customizing the Skin to Modify Global Look and Feel

To customize the skin, you need to understand the structure of the underlying Cascading Style Sheet (CSS). The skin color, font style, or font size can be globally changed in the CSS file.

Customizing Text

Since the mail client supports internationalization, most of the text used in mail are referenced through a special array named `i18n`. The strings are saved in `<uwc-deploy-path>/webmail/en/i18n.js`. To replace the text with a desired text, you need to

look up the string from `<uwc-deploy-path>/webmail/en/i18n.js` and modify it. The default encoding for all files is UTF-8, and you need to ensure that you use the right one, especially when the text contains non-Latin characters.

Customizing Mail by Domain

Mail supports customization specific to each domain. A user who belongs to a specific domain will be directed to a different view (skin) on signing in. Refer to chapter [Chapter 8, “Customizing Domains,”](#) for more information on customizing mail for domain.

Customizing Mail Icons

All Image files are stored in `<uwc-deployed-path>/webmail/ix.` You can look up the image file name from HTML files and replace the URL with the required one.

Adding Extra Features

To add extra features, you can create a new function using the `listFrameHTML()` function in the corresponding `_fs_lr.jsp`. You can also add new functions to be called by `listFrameHTML()`. The new content will then be inserted in the mail application. If the extra feature involves more technical aspects that are not documented, contact Sun Professional Services for assistance.

Understanding the File Layout in Mail

The mail component contains HTML, JavaScript, and JSP files. These files have `.html`, `.js`, and `.jsp` extensions. These files are transmitted to the browser and used to retrieve dynamic contents from the server using a proprietary HTTP protocol. The mail server provides user interfaces for the integrated mail component of Communication Express. The JavaScript files are shared by both these interfaces, and in general provide functionality for the application.

The CSS files that are used in mail are:

- `mail_css_ie5win.css`
- `mail_css_ns6up.css`
- `master-style.css`
- `master-style_ie5up.css`
- `master-style_ns4sol.css`
- `master-style_ns4win.css`
- `master-style_ns6up.css`

[Table 4–1](#) Displays the Different JavaScript Files That are Used in Mail.

TABLE 4-1 JavaScript Files that are Used in Mail

JavaScript file Name	Application
<code>browser.js</code>	Detects the browser to select which .css file should be used
<code>mailui.js</code>	Maintains the common user interface codes shared by messaging listing and Search message result listing
<code>setdomain.js</code>	Checks for Domain security
<code>main.js</code>	Maintains the initialization and globally shared functions
<code>util.js</code>	Utility functions
<code>dtree.js</code>	User interface codes for folder tree

The frame work related file is `en/mail.html`, which is the first file loaded by the browser.

The Mail page is configured to display the Mail page. It creates a number of frames, including several invisible frames. These invisible frames are used to pull the dynamic contents as JavaScript objects from the server. Once downloaded, the hidden frame can invoke the specific callback function in `frameset HTML (_lr_fs.html)` to render the updated dynamic contents.

Mail can be perceived as a dynamic client-side HTML engine. There are not many standard HTML statements in `_lr_fs.html` files. It is the `ListFrameHTML()` function in `_lr_fs.html` files which is responsible for the HTML output. This action takes place when `ListFrameHTML()` accesses the JavaScript objects from the hidden frames such as *cfgFrame*, *msgFrame*, and *mbxFrame* to generate the dynamic HTML page.

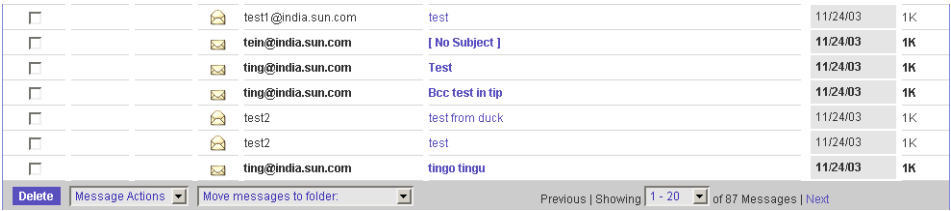
Understanding Basic Interfaces in Mail

This section provides information on the various Mail windows and screens :

- Inbox
- Message Display
- Folder Management
- Compose window
- Message Search window

Inbox

The Inbox screen, shown in [Figure 4-1](#), enables you to view all messages and its basic features. For example, subject, from, received and size. The Inbox screen gets new messages and enables you to search for or delete old messages, as well as move messages into other folders.



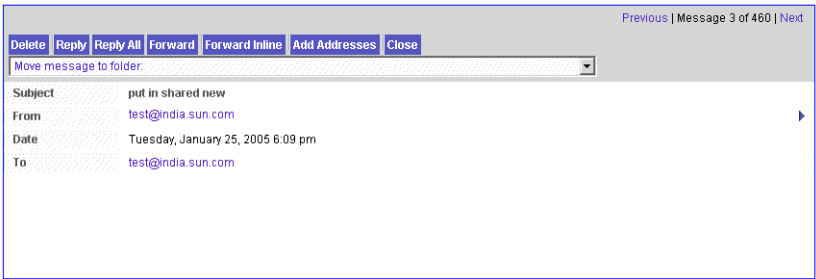
☐			test1@india.sun.com	test	11/24/03	1K
☐			tein@india.sun.com	[No Subject]	11/24/03	1K
☐			ting@india.sun.com	Test	11/24/03	1K
☐			ting@india.sun.com	Bcc test in tip	11/24/03	1K
☐			test2	test from duck	11/24/03	1K
☐			test2	test	11/24/03	1K
☐			ting@india.sun.com	tingo tingu	11/24/03	1K

[Delete](#) | [Message Actions](#) | [Move messages to folder:](#) | [Previous](#) | Showing 1 - 20 of 87 Messages | [Next](#)

FIGURE 4-1 Communications Express Inbox Screen

Message Display

The Message display, shown in Figure 4-2 displays the message selected from the Inbox screen. The Message display gives the option of replying to the sender, forwarding the message, moving the message to a different folder, or deleting the message. The Message screen also enables navigation to the next or previous message.



[Previous](#) | Message 3 of 460 | [Next](#)

[Delete](#) | [Reply](#) | [Reply All](#) | [Forward](#) | [Forward Inline](#) | [Add Addresses](#) | [Close](#)

[Move message to folder:](#)

Subject

put in shared new

From

test@india.sun.com

Date

Tuesday, January 25, 2005 6:09 pm

To

test@india.sun.com

FIGURE 4-2 Message Display

Folder Management

Folder Management, shown in Figure 4-3 displays all folders that can be accessed. The Folder Management screen lists the number of messages contained and the size of each folder. The Folders screen also enables creating new folders, renaming or deleting old ones, subscribing or unsubscribing shared folder, sharing folder, moving a folder within another folder, updating the inbox, and composing new messages. Like the Inbox, the Folders Management also enables collection of external mails.

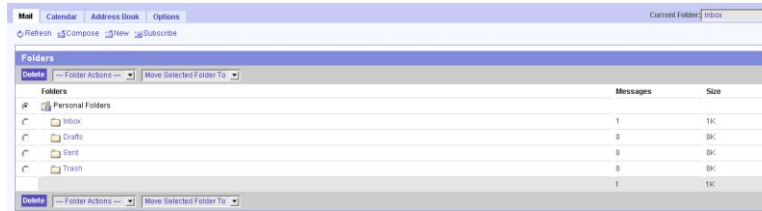


FIGURE 4-3 Folder Management Screen

Compose Window

The Compose window, shown in Figure 4-4, is used to compose a new message. You can also use the window to save a draft or attach a file to the message, look up a recipient in the address book, access the help file, or cancel the composition altogether. Mail recipients can be added in “To”, “Cc”, or “Bcc” fields. You can edit the message in text or HTML format. In the Compose window you can also check the spelling. The Compose window also enables you to set the mail priority or request for a return receipt.

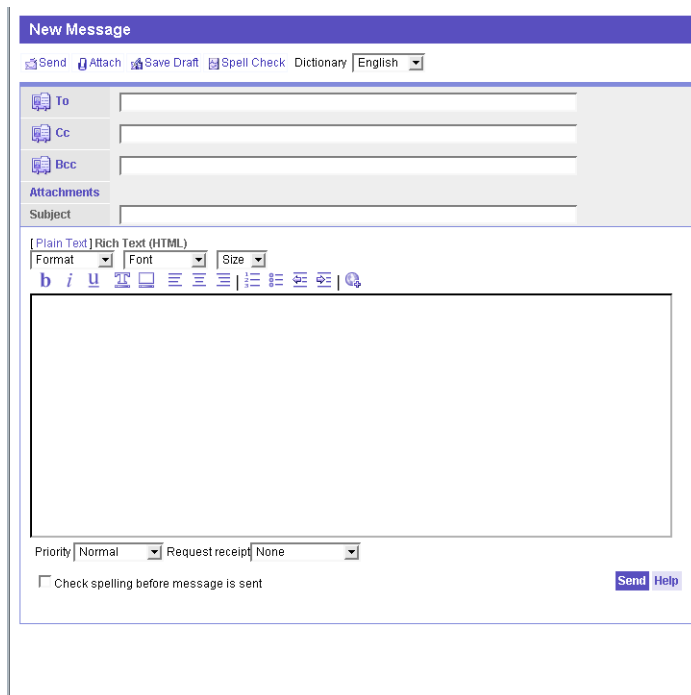
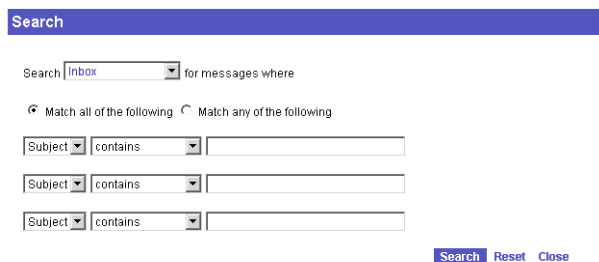


FIGURE 4-4 Compose Window

Message Search Window

The Message Search window, shown in [Figure 4–5](#), is primarily used to search messages by entering sender’s name, subject, body text, or recipient’s name. You can also delete the messages from the search list.



The screenshot shows the Message Search Window. At the top is a purple header bar with the word "Search" in white. Below the header, the text "Search" is followed by a dropdown menu showing "Inbox" and a small downward arrow, followed by the text "for messages where". Below this, there are two radio buttons: the first is selected and labeled "Match all of the following", and the second is labeled "Match any of the following". Below the radio buttons, there are three identical rows of search criteria. Each row consists of a dropdown menu showing "Subject", a dropdown menu showing "contains", and a text input field. At the bottom right of the window, there are three buttons: "Search", "Reset", and "Close".

FIGURE 4–5 Message Search Window

Customizing User Interface Features in Mail

This chapter describes how to customize user interface features in Mail.

This chapter contains the following sections:

- “The Mailbox Tool Bar” on page 59
- “Message Listing” on page 61
- “Customizing The Message Display Window” on page 62
- “The Message Tool Bar” on page 64
- “The Message Composition Window” on page 65
- “Aligning the User Interface from Right to Left (For Arabic only)” on page 66

The Mailbox Tool Bar

This section describes how to modify the mailbox tool bar shown in [Figure 5-1](#).

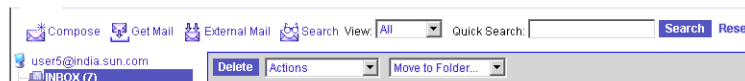


FIGURE 5-1 Mailbox Tool Bar

You can modify the following on the mailbox tool bar:

- Change the layout of the mailbox tool bar relative to the rest of the page
- Rearrange the order of tools
- Change the tools text

Customizing the Mailbox Tool Bar

To modify the mailbox tool bar, edit the appropriate files as follows:

- To customize the layout within the tool bar and the associated graphics, edit the `getToolBar()` function in the `<uwc-deployed-path>/webmail/mbox_fs_lr.jsp` file
- To customize the text associated with the graphics in the Tool Bar, edit the `i18n[ ]` values for get mail, compose, search, new search, file selected message, delete, undelete, and expunge in the `<uwc-deployed-dir>/webmail/<locale>/i18n.js` file

The function `getToolBar()` in `mbox_fs_lr.jsp` handles tool clicks and performs the following tasks:

- Get Mail: *main.refreshMbox()*
- Compose: *main.compose("new")*
- Search: *srch()*

Example: Mailbox Tool Bar Modifications

[Example 5-1](#) shows the necessary edits to be made in the file `mbox_fs_lr.jsp` to :

- Display “Search” as the first tool
- Change the text of “Get Mail” tool to “Get Messages.”

EXAMPLE 5-1 Altered Tool Bar Layout (`mbox_fs_lr.jsp`)

```
function getToolBar() {
    .....
    .....
    .....
    +  '" onclick="Reset()"></span>'+
        nWMtoolbar(i18n['search'], 'srch()',
            'search', 'imx/LrlSearchMsg_wo_1.gif')+
        nWMtoolbar(i18n['compose'], 'main.compose(\'new\')',
            'compose', 'imx/LrlNewMsg_wo_1.gif') +
        nWMtoolbar(i18n['get mail'], SpecialMboxURL('INBOX'),
            'getmail', 'imx/LrlGetMail_wo_1.gif')+
        nWMtoolbar(i18n["collect long"], 'main.collect()',
            'collect', 'imx/LrlColExtMail_wo_1.gif')
}
```

Functionally, the `getToolBar()` function links the “Collect External Mail” hyperlink to `collect()` in `main.js`. [Example 5-2](#) shows how such linking is done.

[Example 5-2](#) shows the necessary changes to be made in file `en/i18n.js` (text)

EXAMPLE 5-2 Altering Tool Bar Text (en/i18n.js)

```
i18n['get mail'] = 'Get Messages'
```

Message Listing

This section describes how to modify the Message List window shown in Figure 5-2.

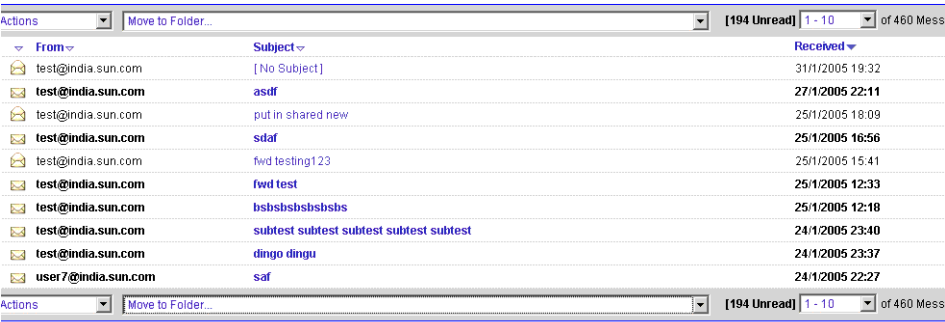


FIGURE 5-2 Message List Window

You can modify the following in the Message List window:

- Change the default sorting order
- Change the text of the default column heading
- Change the text on “Collect External Mail” button

Modifying Message List Window

To modify the Message List window, edit the appropriate files as follows:

- To customize how the message list appears, edit the `listFrameHTML()` function in the `mbox_fs_lr.jsp` file.
- To change the text of default column heading, edit the `i18n[]` values for search results, date, from, to, size, and subject in the `<lang>/i18n.js` file.
- To change the text on the “Collect External Mail” button, edit `i18n[collect long]` in the `<lang>/i18n.js` file.

Functionally, the `getToolBar()` function links the “Collect External Mail” hyperlink to `collect()` in `main.js`. [Example 5–2](#) shows how such linking is done.

Example: Message Listing Modifications

To change the text on “Collect External Mail” button to “Get Messages From Another Server”, follow the code example provided in [Example 5–3](#). This shows the necessary changes to be made in `en/i18n.js` file.

EXAMPLE 5–3 Altering List Window Text (`en/i18n.js`)

```
// POP3 Collection
....
i18n['collect long'] = 'Get Messages From Another Server'
```

Customizing The Message Display Window

This section describes how to modify the Message Display window shown in [Figure 5–3](#).

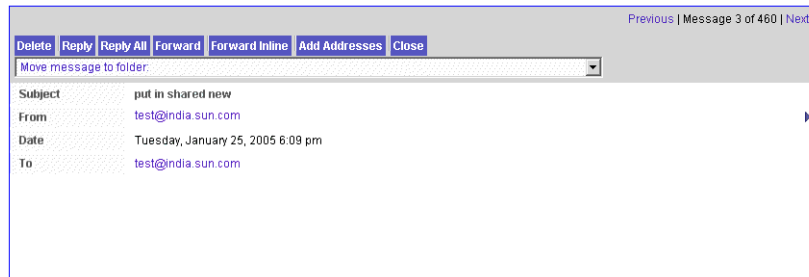


FIGURE 5–3 Message Display Window

You can modify the following in the Message Display window:

- Change the message display
- Alter the layout of the window
- Change the text of the fields
- Display user-defined header fields

Modifying Message Display Window

To modify the Message Display window, edit the appropriate files as follows:

- To customize how the message appears, edit the `listFrameHTML(doc)()` function in the `msg_fs_lr.jsp` file.
- To customize the default text, edit the `i18n[]` values under Message Headers and Message in the `<lang>/i18n.js` file.

Modifying the Message Display Window to Display User Defined Header Fields

This section describes how to add and display user-defined header fields in the Message Display window.

Displaying User Defined Header Fields

[Example 5–4](#) shows how to edit the `listFrameHTML(doc)()` function in the `msg_fs_lr.jsp` file.

Example: Customizing the Message Display Window to Display User Defined Header Fields

This example shows how to display the user defined field *X-document-id* in the Message Display Window.

[Example 5–4](#) shows the edits made to the function `listFrameHTML()` in the `msg_fs_lr.jsp` file.

EXAMPLE 5–4 Changes made to the `msg_fs_lr.jsp` file

```
function listFrameHTML(ftime) {
    ...
    var notifyto
    //add this line here.
    var hdrstr = main.getHeaderStr(main.msgFrame.hdr[0], 'X-document-id')
    ....
    ....
    document.getElementById('POP_UP_DIV').style.display = 'inline'
}
//and add here user defined X-document-id header.
if(hdrstr != '') {
    s += '<tr><td nowrap align=right valign=top width=5%' + main.base_line +
    ' bgcolor=' + main.chrome2 + '>' + main.font() + html('X-document-id') +
    nbsp + '</td>\n<td ' + main.cellBgString + '>' +
    main.font() + hdrstr + '</td></tr>\n'
}
```

EXAMPLE 5-4 Changes made to the `msg_fs_lr.jsp` file (Continued)

```
....  
...  
}
```

The Message Tool Bar

This section describes how to customize the message tool bar.



FIGURE 5-4 Message Tool Bar

You can modify the following in the message tool bar:

- Change the layout of the toolbar relative to the rest of the page
- Alter the layout within the tool bar
- Change the text associated with the graphics

Modifying the Message Tool Bar

To modify the message tool bar, edit the appropriate files as follows:

- To customize the layout within the tool bar, edit the `actionBar()` function in the `msg_fs_lr.jsp` file.
- To customize the texts associated with the graphics in the Tool Bar, edit the `i18n[]` values for compose, reply, reply all, forward, delete, undelete, previous, and next in the `<lang>/i18n.js` file.

Example 5-5 shows the necessary changes to be made in the `en/i18n.js` file.

EXAMPLE 5-5 Altering Message Tool Bar Text (`en/i18n.js`)

```
// Tool Bars  
....  
i18n['previous'] = ' Prev '
```

The Message Composition Window

This section describes how to modify the Compose window.

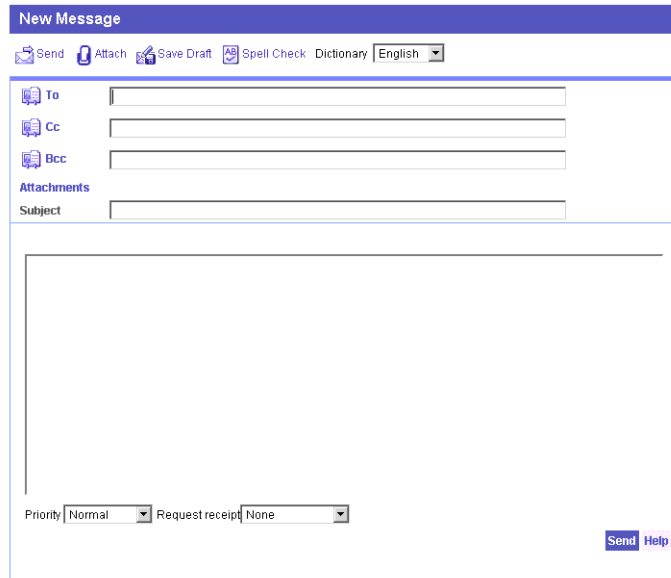


FIGURE 5–5 Compose Window

You can modify the following in the Compose window:

- Change the location of the tools in the window
- Alternative text associated with the tools
- Enable and disable emoticons
- Create your own dictionary

Modifying the Message Composition Window

To modify the Message Composition window, edit the `rtfeditor.html` file.

- To customize the text, edit the values under Message Composition and Tool Bars in the `<lang>/i18n.js` file.
- To enable and disable the emoticons, edit the variable `iconHREF` in the `main.js` file. By default, the emoticon files are located in `<uwc-deployed-path>/webmail/imx` directory.

Functionally, `rtfeditor.html` is called by `comp_fs_lr.html`. This function assembles the code and assigns the functions to the graphics by calling `WMtoolbar()` in `main.js` that also handles colors and text-only versions. The `rtfeditor.html` file generates the “To”, “Cc”, and “Bcc” control area by calling the `i18n_compose_controls()` in `<lang>/i18n.js`.

[Example 5–6](#) shows the necessary changes to be made in the file `en/i18n.js` for changing the text “Recipients” to “Send to”.

EXAMPLE 5–6 Altering Composition Window Text (`en/i18n.js`)

```
// Message Composition
....
i18n['recipient'] = ' Send To '
```

Aligning the User Interface from Right to Left (For Arabic only)

By default, the Menu tabs are aligned from left to right. If you have selected Arabic as the preferred language, then you need to customize Communications Express to align the Menu tabs from right to left.

▼ To Align the User Interface From Right to Left

- To align the User Interface from right to left you need to remove a line from the `ar/i18n.js` file. [Example 5–7](#) shows the line that needs to be added to the `ar/i18n.js`.

Example 5–7 Aligning the User Interface from Right to Left

```
....

    i18n['dir'] = 'rtl'

...
```

Customizing Advanced Features in Mail

This chapter describes how to perform advanced customizations. It also provides HTML files, their location, and details on features that are fully customizable.

This chapter contains the following sections:

- “Files for Customizing the Mail User Interface” on page 67
- “Understanding Shared Folders” on page 70
- “Default LDAP Attributes for Users” on page 70

In addition to the features mentioned in [Chapter 5, “Customizing User Interface Features in Mail”](#), many other features are fully customizable. However, to take advantage of these features, an in-depth knowledge of JavaScript is required. You might also face migration problems when attempting to reconfigure JavaScript files.

Files for Customizing the Mail User Interface

[Table 6–1](#) lists the user interface related FrameSet files of Communications Express user interface that are fully customizable.

TABLE 6–1 Communications Express User Interface Customizable Features

Features	Files
Adds attachment for upload (popup window)	attach_fs_lr.html
Collects pop email (popup window)	collect_fs_lr.html
New message composition	comp_fs_lr.html
Folder viewing and management	fldr_fs_lr.jsp
Message listing for a selected folder	mbox_fs_lr.jsp

TABLE 6-1 Communications Express User Interface Customizable Features (Continued)

Features	Files
Message viewing	msg_fs_lr.jsp
Receipt notification (popup window)	receipt_fs_lr.html
Search message (popup window)	searchmsg_fs_lr.html
Sets folder sharing permission (popup window)	setpermission_fs_lr.html
Search Results	srchresults_fs_lr.html
Subscribe to	subscribe_fs_lr.html

Attachment Options

You can modify the following attachments options in mail:

- Browse button
- Attach, Cancel, Help buttons

HTML File Mapping

The HTML file that controls the attachment related features is `attach_fs_lr.jsp`.

Collect Mail from Another Server

You can modify the following labels that appear when collecting mail from another server:

- POP server name (text field)
- POP user ID (text field)
- Password (text field)
- Delete messages from server (select button)
- Save to folder (list box)
- Collect, Cancel, and Help (button)

HTML File Mapping

The HTML file that controls the mail collection feature is `collect_fs_lr.html`.

Message Composition

The message composition feature enables basic mail functions.

You can modify the following message composition options:

- Compose new message
- Reply to the sender
- Reply to all recipients of the message including the sender
- Forward message to others
- Move message to folder
- Delete message
- Navigate through messages using Prev or Next icon

HTML File Mapping

The HTML file that controls the message composition feature is `msg_fs_lr.html`.

Mailbox Management

The Mail tab enables access to a mailbox.

You can modify the following mailbox management tab options:

- Get new message
- Compose new message
- Search for message
- Move message to selected folder
- Delete message
- Undelete message
- Select message
- Select all messages
- Collect external messages

HTML File Mapping for Mailbox Management

The HTML file that controls the mailbox management tab feature is `mbox_fs_lr.jsp`.

Return Receipt

The return receipt feature enables the management of return receipts.

HTML File Mapping

The file that controls the return receipt feature is `receipt_fs_lr.html`.

Understanding Shared Folders

A shared folder is a folder in which you can store messages that can be accessed by other users. Communications Express manages access to the shared folders by granting others access to the shared folders.

Sharing folders is a two-step process that involves sharing and subscribing.

▼ To Share Folders

- 1 A user shares a folder granting permissions to specific users.
- 2 The users who were given permissions to that folder then subscribe to it.

For more information on Shared Folders, see *Sun Java System Communications Express Online Help*, that can be accessed by clicking the Help button on Communications Express user interface.

Default LDAP Attributes for Users

The Communications Express Server loads a default set of LDAP attributes for a user at the start of a session. These attributes are provided in [Example 6–1](#).

EXAMPLE 6–1 Customizable LDAP attributes

```
cn
givenName
mail
mailAlternateAddress
mailAutoReplyMode
mailAutoReplySubject
mailAutoReplyText
mailAutoReplyTextInternal
mailAutoReplyTimeout
mailDeliveryOption
mailForwardingAddress
mailQuota,mailMsgQuota
preferredLanguage
sn
uid
vacationEndDate
vacationStartDate
```

For more information on the attributes required or allowed by LDAP object classes for Communications Express, see [Sun Java Communications Suite 5 Schema Reference](#)

You might want to obtain other customized LDAP attributes from the server. For example, an Internet Service Provider (ISP) might have a custom LDAP attribute assigned to all users called `myuserclass`. This attribute could denote different types of users that access services, including Communications Express. Possible values for this attribute are `regular` and `vip`.

Note – Before adding custom attributes to the directory, ensure that the directory schema supports these new attributes. The global set of schema for Directory Server can be found in the entry named `cn=schema`.

For more information on extending directory schema, see [Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide](#).

After extending the directory schema, you can add the new attribute to a user or a group entry. To add custom attribute to a user entry you need to modify the entry in `servletsconf` section using the `ldapmodify` command. For more information on `ldapmodify` command, see [Sun Java System Directory Server Enterprise Edition 6.0 Man Page Reference](#).

Depending on the type of user (the value of the `myuserclass` LDAP attribute), different advertisements are presented to the user when they log into Communications Express (Communications Express is customized to display banner advertisement). If the customized client has access to the `myuserclass` LDAP attribute, the type of user can be determined and the relevant banner advertisement for that user type is displayed.

To obtain other customized LDAP attributes from the server, use the `configutil` command to modify the `service.http.extrauserldapattrs` configuration parameter. The attributes are read-only by default. If the user wants to modify an attribute using the Communications Express code, the attribute needs to be marked read-write by appending the suffix.

[Example 6–2](#) assumes the user wants to display banner advertisements depending on the class of the user and that the client program allows the user to edit a link to a home page:

EXAMPLE 6–2 Display of Banner Advertisements Depending on the User Class:

```
configutil -l -o service.http.extrauserldapattrs -v
           "myuserclass,homepage:w "
```


Customizing General Features in Address Book

The Address Book contains many customizable and localizable items that can be changed to create a different look and feel. These customizable items include icons, style sheets, and labels. You can change the position of certain items and also customize captions, headings, button labels, alert, and inline messages.

The rendering of Address Book user interface is handled by XML/XSL files that contain XSL tags, static HTML, and JavaScript. The XSL and JavaScript code are used to display dynamic data and perform actions or form operations. The XSL code in most of the pages is structured into templates which helps make a page modular for various components. All these files are located under the `<uwc-deployed-path>/WEB-INF/ui/html/abs` directory.

This chapter contains information on customizing the following in Address Book:

- “Address Book Icons and Labels” on page 73
- “Style Sheets” on page 75
- “The Main Search Page” on page 78
- “New/Edit Contact Window” on page 79
- “View Contact Window” on page 88
- “New Group or Edit Group Window” on page 89
- “Group Details” on page 89
- “The View Group Window” on page 90
- “Printable Window” on page 90
- “The Import/Export Address Book Window” on page 90

Address Book Icons and Labels

This section discusses how to customize address book icons and labels in Communications Express.

Customizing Icons

All the icons that are used for display in Address Book are defined under two files :

- `search-images.xsl` - Contains most of the icon references that are used in the search result table.
- `commonimages.xsl` - Contains common icon references.

All Address Book related images are located under `<uwc-deployed-path>/WEB-INF/ui/html/abs/`. `Search-images.xsl` and `commonimages.xsl` refer to images in this directory. To customize, you can add or replace the required images in this directory and then change the corresponding references in these files.

[Example 7-1](#) show how you can modify icons.

EXAMPLE 7-1 Changing Icon

To change the icon for Printable as displayed in the toolbar of main page, you need to do the following.

Old Icon: Printable (File name: `- LrlPrintable_1_wo.gif`)

New Icon: New Printable (File name - `ico_print.gif`)

In `commonimages.xsl`, the reference is defined as,

```
<xsl:variable name="print_page.gif" select="'../absimx/LrlPrintable_1_wo.gif'"/>
```

To change the icon file reference,

Add `ico_print.gif` file into `<uwc-deployed-path>/absimx`.

Change the reference in `xsl:variable` tag for `print_page.gif` from `'../absimx/LrlPrintable_1_wo.gif'` to `'../absimx/ico_print.gif'`.

Customizing Labels

All captions and displayable static items that are displayed in the Address Book can be customized. All the labels are defined in the `dictionary-<locale>.xml` file which is located under `<uwc-deployed-path>/WEB-INF/ui/html/abs` directory. This XML file contains definitions in the following form:

```
<word key="_Message">Message Actual Text</word>
```

The key is specified in the XSL file which is used to obtain the value of the `<word>` XML node. This value is eventually displayed during the rendering of XSL into HTML. In the XSL file, the key is specified in the `<xsl:text>` tag.

To customize a particular caption or a static label, change the text corresponding to the desired key. To add new text:

- Add new `<word>` tag with the appropriate key and the prefix `'_'`
- Add the `<xsl:text>` tag with the key as the tag value in the XSL file

[Example 7-2](#) changes the label of the 'New Contact' link on the Main page toolbar to 'Add Contact'. In the example en is assumed to be the language in which the label is displayed.

In `search.xml`, the label is coded as following:

EXAMPLE 7-2 Customization of the Label

```
<a class="ToolLbl" href="javascript:void(0)">
  <xsl:attribute name="onClick">
<xsl:text>javascript:openWinAutoHeight
('addcontact-main.xml?bookid=</xsl:text>
<xsl:value-of select="$selectedBook/entry/@entryID"/>
  <xsl:text>', 'NewContact',
'scrollbars=yes,resizable=yes,width=700');</xsl:text>
</xsl:attribute>
  <xsl:text>_New Contact</xsl:text>      ----> Label Definition
</a>
```

[Example 7-3](#) shows the code before modifying the default definition in `dictionary-en.xml`

EXAMPLE 7-3 Code Before Modification of the Default Definition in Dictionary-en.xml

```
<word key="_View Calendar...">View Calendar...</word>
```

[Example 7-4](#) shows the customized code after modifying the default definition in `dictionary-en.xml`.

EXAMPLE 7-4 Customized Default Definition in Dictionary-en.xml

```
<word key="_View Calendar...">View Schedule...</word>
```

Style Sheets

In Address Book, style sheets play a significant role in the manner in which the user interface is rendered in terms of look and feel as many items that are displayed in Address Book are associated with style sheets. Hence, the look and feel of Address Book pages and pop-ups can be customized by modifying these style sheets.

These style sheets are defined in a CSS file that is included using the `<link>` HTML tag in the HTML `<head>` section of the XSL files. These style sheets are referenced by assigning a 'class' attribute of HTML tags.

[Example 7-5](#) shows the class attribute of HTML tags.

EXAMPLE 7-5 Class attribute of HTML tags

```
<td class="TdActTdLst">.....</td>
```

[Example 7-6](#) shows the class attribute of XSL tags.

EXAMPLE 7-6 Class attribute of XSL tags

```
<xsl:variable name="css-class" select="'TdActTdLst'"/>
....
<td>
  <xsl:attribute name="class">
    <xsl:value-of select="$css-class"/>
  </xsl:attribute>
  ...
</td>
```

The Address Book has two separate CSS files for Internet Explorer and Netscape. These files exist in `<uwc-deployed-path>/absimx` and are named as:

- `ab_css_ie5win.css` for Internet Explorer
- `ab_css_ns6up.css` for Netscape

What can be Customized in Style Sheets

- You can change the definition of an existing style sheet but, you need not change the class attribute reference.
- You can add new style sheet definitions and change the class attribute reference for the desired HTML tag in XSL files.

Changing the Definitions of an Existing Style Sheet

The definition of a primary button display is `Btn1` and is defined in the CSS file as shown in [Example 7-7](#).

EXAMPLE 7-7 Before Changing the Definition of a Primary Button Display "Btn1"

```
.Btn1 {background:#594fbf;color:
      #FFF;font-weight:bold;padding:2px 0px;margin:
0px 0px 1px 0px;border:0px none #000; }
```

For more information on how to use the class in the XSL file, see [“Displaying User Defined Header Fields” on page 63](#). [Example 7-8](#) shows how the class is used in the XSL file for the “Close” button.

EXAMPLE 7-8 Class Used in the XSL File for the “Close” Button

```
<input class="Btn1" id="close" onBlur="if (this.disabled==0)
this.className='Btn1'" onMouseover="if (this.disabled==0)
this.className='Btn1Hov'" onFocus="if (this.disabled==0)
this.className='Btn1Hov'" onClick="parent.close()" tabIndex="1"
onMouseout="if (this.disabled==0) this.className='Btn1'"
type="button" name="close" value="Close">
```

[Example 7-9](#) shows the code after changing the definition of a primary button display.

EXAMPLE 7-9 After Changing the Definition of a Primary Button Display "Btn1"

```
Btn1 {
    background:#ffffff;color:#000;
    font-weight:bold;padding:2px 0px;
    margin:0px 1px 1px 0px;
    border:0px none #000;
}
```

Adding New Style Sheet Definitions

You can add new style sheet definitions and also change the class attribute reference for the desired HTML tag in XSL files. You can add a different class definition altogether in the CSS file, where the newly defined class needs to be assigned to the XSL file.

```
.EMPBtn { font-family: Arial, Helvetica, sans-serif; text-decoration none;
font-weight:bold; color: #594fbf; }
```

[Example 7-10](#) shows the corresponding changes in the XSL file.

EXAMPLE 7-10 After Changing Class Attribute Reference for the Desired HTML Tag

```
<input class="EMPBtn" id="close"onBlur="if(this.disabled==0)
this className='Btn1'"onMouseover="if(this.disabled==0)
this.className='Btn1Hov'"onFocus="if(this.disabled==0)
```

EXAMPLE 7-10 After Changing Class Attribute Reference for the Desired HTML Tag (Continued)

```
this.className='Btn1Hov'"onclick="parent.close()"tabIndex"1"
onmouseout="if (this.disabled==0)
this.className='Btn1'"type="button"name="close"value="Close">
```

The Main Search Page

The search page can be divided into following sections:

- Tabs
- Toolbar Pane
- Search Pane
- Action Row
- Search Result Table
- Quick Add Pane

Each of these sections are defined in templates that exist either in search.xml or search-template.xml. The search-template.xml file contains commonly used templates shared by the Main Search Page, Search Address Book pop-up, and Add Addresses pop-up.

These sections can be reorganized by modifying their respective positions. Table 7-1 shows the commonly used templates for search-template.xml and search.xml

TABLE 7-1 Commonly used templates in search-template.xml and search.xml

Sections	Template Name	File
Tabs	search-template-bookbar	search-template.xml
Toolbar	search-template-toolbar	search.xml
Search Pane	search-template-searchbar	search-template.xml
Action Row	search-template-actionbar	search.xml
Search Result Table	search-template-searchresult	search-template.xml
Quick Add	quick-add	search.xml

Search Result Table

The templates that refer to Search Result Table are:

- **commonattributes-list.xml**. The templates defined in this file are used in the display of header captions in result table in a particular layout.
- **entries-list.xml**. The templates defined in this file are used to display the actual data such as email, displayname, web address and so on in a particular format.

To customize the Search Main Page, Search Address Book popup, Add Addresses, and Printable views, you need to change the layout information in these templates.

New/Edit Contact Window

The files pertaining to the new contact or edit contact window are:

- New contact - addcontact.xml
- Edit contact - editcontact.xml

Common code shared among new and edit window can be accessed from common-editcontact.xml

Table 7-2 lists the templates that are defined for the sections in New Contact or Edit Contact Window

TABLE 7-2 Templates Applicable for New/Edit Contact Window

Sections	Template Name	File
Skip Links	-	addcontact.xml and editcontact.xml
Name and Company	nameAndCompany	common-editcontact.xml
Phone	phone	common-editcontact.xml
Email	email	common-editcontact.xml
Addresses	address	common-editcontact.xml
IM Nicknames	IM	common-editcontact.xml
Web Addresses	onlineInfos	common-editcontact.xml
Notes	notes	common-editcontact.xml

You can customize the window by changing labels or reordering the listed sections in any desired order.

In `addcontact.xml`, the 'edit-abperson' template calls the templates.

[Example 7-11](#) shows the code that displays Address before Online information.

EXAMPLE 7-11 Code that Displays Address Before Online Information

```
<xsl:call-template name="address">
  <xsl:with-param name="abperson" select="$abperson"/>
</xsl:call-template>
<xsl:call-template name="onlineInfos">
  <xsl:with-param name="abperson" select="$abperson"/>
</xsl:call-template>
```

[Example 7-12](#) shows how to display Online information before Address.

EXAMPLE 7-12 Code that Displays Online Information Before Address

```
<xsl:call-template name="onlineInfos">
  <xsl:with-param name="abperson" select="$abperson"/>
</xsl:call-template>

<xsl:call-template name="address">
  <xsl:with-param name="abperson" select="$abperson"/>
</xsl:call-template>
```

Phone

Address Book allows you to set five types of phones that are defined in `phoneEmailAndIM`. The five definitions of 'phone' template are called and the priority value is passed to them when the values are displayed in the drop-down list. You can change the priority orders.

Phone

Phone (1):
 Work

Phone (2):
 Home

Phone (3):
 Mobile

Phone (4):
 Pager

Phone (5):
 Fax

[Back to Top](#)

FIGURE 7-1 Customizing Phone

[Example 7-13](#) displays the default code where Work phone is displayed as priority 1 and Home phone is displayed as priority 2.

EXAMPLE 7-13 Default Code that Displays Work Phone as Priority 1 and Home Phone as Priority 2

```

<option value="work">
  <xsl:choose>
    <xsl:when test="$abperson">
      <xsl:choose>
        <xsl:when test=
          "$abperson/phone[@priority = $priority]">
          <xsl:if test=
            "$abperson/phone[@priority = $priority]/@type = 'work'">
            <xsl:attribute name="selected"/>
          </xsl:if>
        </xsl:when>
        <xsl:otherwise>
          <xsl:if test="$priority = '1'">
            <xsl:attribute name="selected"/>
          </xsl:if>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <xsl:if test="$priority = '1'">
        <xsl:attribute name="selected"/>
      </xsl:if>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>_Work</xsl:text>
</option>
<option value="home">
  <xsl:choose>
    <xsl:when test="$abperson">
      <xsl:choose>
        <xsl:when test=
          "$abperson/phone[@priority = $priority]">
          <xsl:if test=

```

EXAMPLE 7-13 Default Code that Displays Work Phone as Priority 1 and Home Phone as Priority 2
(Continued)

```
        "$abperson/phone[@priority =
        $priority]/@type = 'home'">
        <xsl:attribute name="selected"/>
    </xsl:if>
</xsl:when>
<xsl:otherwise>
    <xsl:if test="$priority = '2'">
        <xsl:attribute name="selected"/>
    </xsl:if>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
    <xsl:if test="$priority = '2'">
        <xsl:attribute name="selected"/>
    </xsl:if>
</xsl:otherwise>
</xsl:choose>
<xsl:text>_Home</xsl:text>
</option>
```

[Example 7-14](#) shows how to change the code so that it displays Home phone as priority 1 and Work phone as priority 2.

EXAMPLE 7-14 Code that Displays Home Phone as Priority 1 and Work Phone as Priority 2

```
<option value="work">
    <xsl:choose>
        <xsl:when test="$abperson">
            <xsl:choose>
                <xsl:when
                    test="$abperson/phone[@priority = $priority]">
                    <xsl:if test=
"$abperson/phone[@priority = $priority]/@type = 'work'">
                        <xsl:attribute name="selected"/>
                    </xsl:if>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:if test="$priority = '2'">
                        <xsl:attribute name="selected"/>
                    </xsl:if>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:when>
    </xsl:choose>
</option>
```

EXAMPLE 7-14 Code that Displays Home Phone as Priority 1 and Work Phone as Priority 2 (Continued)

```

        </xsl:choose>
      </xsl:when>
      <xsl:otherwise>
        <xsl:if test="$priority = '2'">
          <xsl:attribute name="selected"/>
        </xsl:if>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:text>_Work</xsl:text>
  </option>
  <option value="home">
    <xsl:choose>
      <xsl:when test="$abperson">
        <xsl:choose>
          <xsl:when test="$abperson/phone[@priority = $priority]">
            <xsl:if
test="$abperson/phone[@priority = $priority]/@type = 'home'">
              <xsl:attribute name="selected"/>
            </xsl:if>
          </xsl:when>
          <xsl:otherwise>
            <xsl:if test="$priority = '1'">
              <xsl:attribute name="selected"/>
            </xsl:if>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:when>
      <xsl:otherwise>
        <xsl:if test="$priority = '1'">
          <xsl:attribute name="selected"/>
        </xsl:if>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:text>_Home</xsl:text>
  </option>

```

Email

Address Book allows you to set three types of email that are defined in `phoneEmailAndIM`. The three definitions of email templates are called and the priority values are passed to them. You can change the priority orders.

Email

Email (1): Work ☐	Email (3): Pager ☐
Email (2): Home ☐	

[Back to Top](#)

FIGURE 7-2 Customizing Email

Dates

Address Book allows you to set Birthday and Anniversary dates. These two dates are defined in the 'importantDates' template. This template passes all relevant information to the 'date' template.

You can customize the labels, change layout such as size of fields for the dates or their respective positions.

EXAMPLE 7-15 Default Code that Shows Horizontally Aligned Fields and Displays Birthday as the First Field and Anniversary as the Second

```
<tr>
  <xsl:call-template name="date">
    <xsl:with-param name="abperson" select="$abperson"/>
    <xsl:with-param name="type" select="'birthday'"/>
    <xsl:with-param name="index" select="'1'"/>
    <xsl:with-param name="dateFormat" select="$dateFormat"/>
    <xsl:with-param name="width" select="40"/>
    <xsl:with-param name="fieldName" >
      <xsl:value-of select="$fieldPrefix" />
      <xsl:text>dateOfBirth</xsl:text>
    </xsl:with-param>
    <xsl:with-param name="label">
      <xsl:text>_ Birthday</xsl:text>
    </xsl:with-param>
    <xsl:with-param name="tabindex" select="'52'"/>
  </xsl:call-template>

  <xsl:call-template name="date">
    <xsl:with-param name="abperson" select="$abperson"/>
    <xsl:with-param name="type" select="'anniversary'"/>
    <xsl:with-param name="index" select="'2'"/>
    <xsl:with-param name="dateFormat" select="$dateFormat"/>
    <xsl:with-param name="width" select="65"/>
    <xsl:with-param name="fieldName" >
```

EXAMPLE 7-15 Default Code that Shows Horizontally Aligned Fields and Displays Birthday as the First Field and Anniversary as the Second (Continued)

```

        <xsl:value-of select="$fieldPrefix" />
        <xsl:text>anniversary</xsl:text>
      </xsl:with-param>
      <xsl:with-param name="label">
        <xsl:text>_ Anniversary</xsl:text>
      </xsl:with-param>
      <xsl:with-param name="tabindex" select="'55'"/>
    </xsl:call-template>
  </tr>

```

[Example 7-16](#) shows how to display the fields in a vertical layout and also rearranges Anniversary as the first date and Birthday as second

EXAMPLE 7-16 Code that Displays the Field in a Vertical Layout and also Rearranges Anniversary as the First Date and Birthday as Second

```

<tr>
  <xsl:call-template name="date">
    <xsl:with-param name="abperson" select="$abperson"/>
    <xsl:with-param name="type" select="'anniversary'"/>
    <xsl:with-param name="index" select="'1'"/>
    <xsl:with-param name="dateFormat" select="$dateFormat"/>
    <xsl:with-param name="width" select="65"/>
    <xsl:with-param name="fieldName" >
      <xsl:value-of select="$fieldPrefix" />
      <xsl:text>anniversary</xsl:text>
    </xsl:with-param>
    <xsl:with-param name="label">
      <xsl:text>_ Anniversary</xsl:text>
    </xsl:with-param>
    <xsl:with-param name="tabindex" select="'52'"/>
  </xsl:call-template>
</tr>
<tr>
  <xsl:call-template name="date">
    <xsl:with-param name="abperson" select="$abperson"/>
    <xsl:with-param name="type" select="'birthday'"/>
    <xsl:with-param name="index" select="'2'"/>
    <xsl:with-param name="dateFormat" select="$dateFormat"/>
    <xsl:with-param name="width" select="40"/>
    <xsl:with-param name="fieldName" >
      <xsl:value-of select="$fieldPrefix" />
      <xsl:text>dateOfBirth</xsl:text>
    </xsl:with-param>
  </xsl:call-template>
</tr>

```

EXAMPLE 7-16 Code that Displays the Field in a Vertical Layout and also Rearranges Anniversary as the First Date and Birthday as Second (Continued)

```
</xsl:with-param>
<xsl:with-param name="label">
  <xsl:text>_Birthday</xsl:text>
</xsl:with-param>
<xsl:with-param name="tabindex" select="'55'"/>
</xsl:call-template>
```

Web Addresses

Address Book allows you to set four web addresses for work, home, calendar, and freebusy. These addresses are defined in the *'onlineInfos'* template.

The *'onlineInfos'* template calls the following:

- *'weburl'* that passes the priority for ordered display
- *'calendar'* with *type="calendar"* for calendar URL and *type="freebusy"* for *freebusyurl"*

You can change the order in which these addresses appear in a way similar to the examples illustrated above.

IM Nicknames

Address Book allows you to set two IM nicknames. The list of IM services that is supported by Address Book are listed in the drop-down menus. These two IM options are defined in *'phoneEmailAndIM'* template which calls the *'im'* template passing the relevant information.

- You may change the order of IMs by changing their priorities as illustrated in previous examples.
- You may also change the default selected services. The default services available in the drop down list are *'SunONE'* and *'AIM'*. The code checks for the passed priority and marks the appropriate option in the drop down as selected. You may shuffle the priority value in the code to select a different service. This code exists in the *'im'* template.

[Example 7-17](#) shows the default code that displays SunONE as the default selected service

EXAMPLE 7-17 Default Code that Displays SunONE as the Default Service

```
<option value="SunONE">
  <xsl:choose>
    <xsl:when test=
```

EXAMPLE 7-17 Default Code that Displays SunONE as the Default Service *(Continued)*

```

"$abperson and $abperson/im[@priority=$priority]">
  <xsl:if test=
    "$abperson/im[@priority=$priority]/@service = 'SunONE'">
    <xsl:attribute name="selected"/>
  </xsl:if>
</xsl:when>
<xsl:otherwise>
  <xsl:if test="$priority= '1'">
---- This value is passed by 'phoneEmailAndIM'
---template to have this option default selected
    <xsl:attribute name="selected"/>
  </xsl:if>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>iPlanet</xsl:text>
</option>
.....
<option value="Yahoo">
  <xsl:choose>
    <xsl:when
test="$abperson and $abperson/im[@priority=$priority]">
      <xsl:if
test="$abperson/im[@priority=$priority]/@service = 'Yahoo'">
        <xsl:attribute name="selected"/>
      </xsl:if>
    </xsl:when>
    <xsl:otherwise>
      <xsl:if test="$priority = '3'">
        <xsl:attribute name="selected"/>
      </xsl:if>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:text>Yahoo</xsl:text>
  </option>

```

Example 7-18 shows the modified code that displays Yahoo as the default Service

EXAMPLE 7-18 Modified Code that Displays Yahoo as the Default Service

```

<option value="SunONE">
  <xsl:choose>
    <xsl:when
test="$abperson and $abperson/im[@priority=$priority]">

```

EXAMPLE 7-18 Modified Code that Displays Yahoo as the Default Service *(Continued)*

```
<xsl:if
test="$abperson/im[@priority=$priority]/@service = 'SunONE'">
  <xsl:attribute name="selected"/>
</xsl:if>
</xsl:when>
<xsl:otherwise>
  <xsl:if test="$priority= '3'">
---- This value is passed by 'phoneEmailAndIM'
----template to have this option default selected
    <xsl:attribute name="selected"/>
  </xsl:if>
  </xsl:otherwise>
</xsl:choose>
<xsl:text>iPlanet</xsl:text>
</option>
.....
<option value="Yahoo">
  <xsl:choose>
    <xsl:when
test="$abperson and $abperson/im[@priority=$priority]">
      <xsl:if
test="$abperson/im[@priority=$priority]/@service = 'Yahoo'">
        <xsl:attribute name="selected"/>
      </xsl:if>
    </xsl:when>
    <xsl:otherwise>
      <xsl:if test="$priority = '1'">
        <xsl:attribute name="selected"/>
      </xsl:if>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:text>Yahoo</xsl:text>
  </option>
```

View Contact Window

The View contact window displays the sections in the same order as defined in new/edit contact window. You may customize this window by modifying labels, reordering sections, modifying layout, alignment in the page, and so on. The file to be changed for this window is 'viewcontact.xml'

New Group or Edit Group Window

The files pertaining to the New Group or Edit Group window are:

- New Group - `addgroup.xml`
- Edit Group - `editgroup.xml`

The common code shared among new and edit contact are stored in `common-editgroup.xml`.

[Table 7–3](#) lists the templates defined for the various sections in New Group or Edit group

TABLE 7–3 Templates Defined for Various Sections in the New/Edit group.

Sections	Template Name	File
Group Details	groupDetails	common-editgroup.xml
Group Members	groupMembers	common-editgroup.xml

You can customize the window by changing labels or reordering the listed sections in a desired format.

Group Details

The Group Details section displays the following fields:

- Display name
- Description
- Web URL
- Calendar URL

You can reorder the positions of these fields. You can also add the email address which is supported for a group.

[Example 7–19](#) displays the code to be used for adding the email address which is supported for a group.

EXAMPLE 7–19 Code to be Used for Adding the Email Address that is Supported for a Group

```
<tr>
  <td class="PropLabelAb" width="22%"><span class="Lbl2">
<label for="email"><xsl:text>_Email</xsl:text></label></span></td>
  <td class="PropInput" width="55%">
    <input type="text" size="30"
maxlength="30" name="{ $fieldPrefix }piEmail1" id="email"
tabindex="2" onChange="javascript:this.value=this.value.trim()"
```

EXAMPLE 7-19 Code to be Used for Adding the Email Address that is Supported for a Group
(Continued)

```
onBlur="javascript:this.value=this.value.trim() ">
    <xsl:attribute name="value">
      <xsl:value-of
select="/xslui/iab/bookentry/group/entry/email"/>
      </xsl:attribute>
    </input>
  </td>
  <td width="6%">&nbsp;</td>
  <td width="11%">&nbsp;</td>
  <td width="6%">&nbsp;</td>
</tr>
```

The View Group Window

The View Group window displays the sections in the same order as defined in New Group or Edit Group window. You may customize this window by modifying labels, reordering sections, modifying layout, their alignment in the page. The file to be modified for this window is 'viewgroup.xsl'

Printable Window

The file that is used for modifying the Printable window is `abprnlist.xsl`. The window displays each contact/group in card format. You can customize the page by changing the layout or changing the order of fields and stylesheet items displayed in the page.

The Import/Export Address Book Window

The file that is used for modifying the import/export window is `importexport.xsl`. The page displays Import and Export sections.

Import and Export Address Book

*Indicates a required field

Import

* Import from File:

Import Format:

Export

Export Format:

FIGURE 7-3 Customizing the Import/Export Address Book Window.

You can customize the page for:

- Labels
- Swapping the Import and Export sections
- Change the format options displayed in drop-downs
- Layout

For example, let us consider modifying the listed format options. The format options listed are:

- Netscape Communication Address Book (LDIF)
- Microsoft Outlook CSV
- Address Book CSV
- Mozilla Thunderbird CSV
- vCard

Example 7-20 shows the code before modifying the options list.

EXAMPLE 7-20 Code Before Modifying the Options List

```
<td class="PropInput" width="80%">
  <select name="importformat" tabindex="2">
    <option value="ldif"><xsl:text>_Netscape Ldif</xsl:text></option>
    <option value="csvus"><xsl:text>_Outlook CSV</xsl:text></option>
    <option value="iabs"><xsl:text>_SunONE CSV</xsl:text></option>
    <option value="iabs"><xsl:text>_Thunderbird CSV</xsl:text></option>
```

EXAMPLE 7-20 Code Before Modifying the Options List *(Continued)*

```
<option value="vcard"><xsl:text>_vCard</xsl:text></option>
</select>
</td>
```

The options list can be modified to display in the following order in drop down list.

- Address Book CSV
- Microsoft Outlook CSV
- Netscape Communication Address Book (LDIF)
- Mozilla Thunderbird CSV
- vCard

[Example 7-21](#) shows the code after modifying the options list

EXAMPLE 7-21 Code After Modifying the Options List

```
<td class="PropInput" width="80%">
<select name="importformat" tabindex="2">
<option value="iabs"><xsl:text>_SunONE CSV</xsl:text></option>
<option value="csvus"><xsl:text>_Outlook CSV</xsl:text></option>
<option value="ldif"><xsl:text>_Netscape Ldif</xsl:text></option>
<option value="ldif"><xsl:text>_Thunderbird CSV</xsl:text></option>
<option value="vcard"><xsl:text>_vCard</xsl:text></option>
</select>
</td>
```

Customizing Domains

Communications Express allows you to customize Mail, Calendar, and Address Book on a specific domain.

This chapter has the following contents:

- “Default User Preferences” on page 93
- “Themes for a Specific Domain and Locale” on page 94
- “Mail Customization for a Specific Domain” on page 95
- “Address Book Customization for a Specific Domain and a Locale” on page 96

Default User Preferences

The default user preferences that is to be used for a first-time user can be configured on a per-domain basis. The default user preferences can be configured in the `uwcdomainconfig.properties` file.

▼ To Customize the Default User Preferences for a Domain

To configure default user preferences for a specific domain, for example, `siroe.com`, you need to do the following:

- 1 **Create a directory with the name same as the domain name in `<domain-dir>` directory.**

For example, create `<domain-dir>/siroe.com` directory.

- 2 **Create a directory with the locale name inside the domain directory.**

For example, if you want to have English locale for `siroe.com`, you must create a folder `en` in the domain directory. The directory structure would look as follows:

`/<domain-dir>/siroe.com/en`

- 3 **Copy** `uwcdomainconfig.properties` **file from** `<domain-dir>` **directory to the newly created directory above.**
- 4 **Make the required changes to** `uwcdomainconfig.properties` **file in** `<domain-dir>/siroe.com` **directory.**
You can configure the default user preferences for global, calendar and address book.
- 5 **Restart the web server to apply changes**

Themes for a Specific Domain and Locale

You can configure themes on a per-domain basis.

Note – Currently Communications Express does not support customization of themes on a per-user basis

▼ To Customize a Theme for a Specific Domain

To configure a theme for a specific domain, for example, `siroe.com` and a locale `ja` and `fr`, you need to do the following.

- 1 **Create a directory with the name same as the domain name in** `<domain-dir>` **directory. For example, create** `<domain-dir>/siroe.com` **directory**
- 2 **Copy** `uwcdomainconfig.properties` **file from** `<domain-dir>` **to the** `<domain-dir>/siroe.com` **directory.**
- 3 **Modify the parameter** `uwc-user-attr-SunUCTheme` **to the required theme name in** `uwcdomainconfig.properties` **file in** `<domain-dir>/siroe.com` **directory.**

For example, to configure a theme “christmas” add the parameter `uwc-user-attr-SunUCTheme-<locale-name>=christmas`, where `<locale-name>` is the name of the locale for which you want the theme. For example:
`uwc-user-attr-SunUCTheme-en=christmas` would correspond to the English locale. To configure a theme called “christmasJA” for Japanese locale, set `uwc-user-attr-SunUCTheme-ja=christmasJA`. To configure a theme called “christmasFR” for french locale, set `uwc-user-attr-SunUCTheme-fr=christmasFR`. The parameter “`uwc-user-attr-SunUCTheme`” is used for default locale. It is important that though the theme name could be anything, the property name should end with corresponding locale as mentioned in the example.

- 4 **Create the directories** `<skin-dir>/christmasJA`, `<skin-dir>/christmasFR` and `<skin-dir>/christmas`.
- 5 **Copy** `themes.properties` **from** `<skin-dir>/` **to** `<skin-dir>/christmasJA`, `<skin-dir>/christmasFR` and `<skin-dir>/christmas`.
- 6 **Edit the corresponding** `themes.properties` **file to point to the customized style sheets and images**.
- 7 **Restart the web container on which Communications Express is deployed to apply changes**.

Mail Customization for a Specific Domain

This section describes how to customize the mail interface for each domain.

You can perform the following tasks to customize the client interface:

- Create a directory for mail with the domain name under `<uwc-deployed-dir>/webmail/` directory.
- Copy the files from the default directory in the same hierarchy and customize them.
For example, assume that you have a domain called `siroe.com`. To change the icon for the domain `siroe.com`, add a new icon in the `imx` directory of `siroe.com` and change the reference to it in the `main.js` file. [Table 8–1](#) shows the directory structure for the domain `siroe.com`

TABLE 8–1 Directory Structure for the Domain `siroe.com`.

Directory	Description
<code>webmail/...</code>	// default interface
<code>webmail/imx/...</code>	// default image
<code>webmail/en/...</code>	// default language
<code>webmail/siroe.com/main.js</code>	// refers to the domain specific file
<code>webmail/siroe.com/imx/bottle.gif</code>	// refers to the customized images

- After login, the server refers the user agent to pick `themail.html` file that is located in the `domain/lang` directory. The `mail.html` file contains the relative references to the rest of the interface. The client requests all the files in the directory to make the interface. If these files exist in the `domain/lang` directory they are displayed, otherwise the default setup files from `<uwc-deployed-dir>webmail/en/` are displayed.
If you have many domains and only a few distinct "brands" then you can use links to make the server point to the correct brand

TABLE 8-2 Linking Multiple Domains to Few Distinct Brands

Directory	Description
html/...	// default interface
html/sesta.com/...	// customized interface for brand 1
html/varrius.com -> sesta.com	// default interface

Domain Customization Based on URL

The server listens to all IP addresses and presents a customized interface before the authentication occurs. The server does this by looking at the URL and by determining if it contains a known domain and presents the per domain Login screen for the domain.

For example, for the per domain Login screen: `http://webmail.sesta.com/`, the server presents the page from the location:
`<uwc-deployed-dir>/webmail/<domain-name>/en/default.html`.

In this case a user does not have to suffix `@domain` to the user name to login.

For more information on Customizing the Mail Component, Sun Java System Messenger Express Customization Guide.

Address Book Customization for a Specific Domain and a Locale

In address book, all user interface related source files (XML/XSL) are located under `<uwc-deployed-path>/WEB-INF/ui/html/abs` directory by default. To customize the user interface for a domain, follow these steps.

▼ To Customize Address Book for a Specific Domain and Locale

- 1 **Create a directory with a domain name, (For example siroe.com) under** `<uwc-deployed-path>/WEB-INF/ui/html`.
- 2 **Copy all files from** `<uwc-deployed-path>/WEB-INF/ui/html/abs` **to** `<uwc-deployed-path>/WEB-INF/ui/html/<domain>`.

3 Make the required changes in XSL files as explained in the chapter [Chapter 7, “Customizing General Features in Address Book.”](#)

Although all images are located under `<uwc-deployed-path>/absimx` by default, you may create a directory with a domain name (For example `<uwc-deployed-path>/absimx/siroe.com`) under it and include the icons that are customizable under the newly created domain directory. Also ensure that the path reference to the newly created directory is updated in `<uwc-deployed-path>/WEB-INF/ui/html/<domain> search-images.xsl` and `commonimages.xsl`.

Localizing Communications Express

Communications Express allows you to localize Mail, Calendar, Address Book, and Options as per your requirements.

- [“Calendar and Options” on page 99](#)
- [“Languages that use Characters Outside the ASCII Range” on page 101](#)
- [“Mail” on page 101](#)
- [“Adding Locales” on page 102](#)
- [“Address Book” on page 104](#)

Calendar and Options

To localize text, you need to change the `i18n.properties` file that is located in `<domain-dir>/<locale>` directory. If the deployment is configured to work with multiple domains and multiple locales, then a directory by that domain name is created under `<domain-dir>`

For example, if there is a domain called `example.com`, perform the following steps.

- Create `<domain-dir>/example.com`.
- Create directories for each supported locale (`en_US`, `ja`, `de_DE`, `fr_FR`, and so on) under that domain, copy `i18n.properties` to the domain, and change it accordingly.
- Copy `<domain-dir>/<locale>/i18n.properties` to `<domain-dir>/example.com/<locale>` directory.

Date/Time Formats for Calendar

The localization of the Date/Time formats for Calendar should be performed by constructing locale-specific patterns for certain symbols, delimiters and hard-coded strings that are enclosed in single quotes. These symbols are converted into locale specific strings by Communications Express.

For more information on localizing date and time formats in calendar, see [“Date/Time Formats for Calendar” on page 99](#).

TABLE 9-1 Localizing Date/Time Formats for Calendar

Symbol	Meaning	Presentation	Example
y	year	Number	1996
M	month in year	Text&Number	July & 07
d	day in month	Number	10
h	hour in am/pm (1-12)	Number	12
H	hour in day (0-23)	Number	0
m	minute in hour	Number	30
s	second in minute	Number	55
S	millisecond	Number	978
E	day in week	Text	Tuesday
D	day in year	Number	189
F	day of week in month	Number	2 (2nd Wed in July)
w	week in year	Number	27
W	week in month	Number	2
a	am/pm marker	Text	PM
z	time zone	Text	Pacific Standard Time
”	escape for text	Delimiter	(none)
”	single quote	Literal	.

Characters that are displayed in the table above are treated as quoted text. These characters appear in the formatted text even if they are not enclosed within single quotes.

The number of symbol letters you specify also determines the format. For example, if the “zz” pattern results in “PDT,” then the “zzzz” pattern generates “Pacific Daylight Time.” [“Date/Time Formats for Calendar” on page 99](#) summarizes these rules.

TABLE 9-2 Localization Rules

Presentation	Number of Symbols	Result
Text	1-3	”abbreviated form, if one exists

TABLE 9-2 Localization Rules (Continued)

Presentation	Number of Symbols	Result
Text	≥ 4	full form
Number	minimum number of digits is required	shorter number are padded with zeros (for year, if the count of "y" is 2, then the year is truncated to 2 digits)
Text & Number	1-2	text form
Text & Number	3	number form

Languages that use Characters Outside the ASCII Range

You may want to use the `native2ascii` utility provided by JDK to convert non-ascii characters into escaped unicode values. This utility is available at `<jdk-install-dir>/bin`

To use the `native2ascii` utility, enter non-ascii characters for the corresponding values in the properties file and save it.

Use the command, `native2ascii -encoding <native_encoding> <src_filename> <target_filename>`

where,

`native_encoding` represents the encoding used to create the source file, `src_filename` represents the source file and `target_filename` represents the `i18n.properties` file.

Mail

Communications Express allows you to localize any feature of Mail. Different pages can be created for different languages. When you create language-specific static pages, you need to group them in subdirectories under the main document directory. The code automatically detects the client's language preference and fetches the pages from the appropriate subdirectory.

When you change common sections for static pages, you must make the changes in multiple locale (languages) directories, if modifications are desired across languages (for example, if changes are required to modify JavaScript behavior). Conversely, you can also make language-specific modifications selectively throughout the application.

Specific Locales

[“Mail” on page 101](#) lists the specific locales and their abbreviations that are supported by Communications Express services. The default language is English.

TABLE 9-3 Communications Express Specific Locales

Locale	Abbreviation
English	en
Japanese	ja
Spanish	es
French	fr
German	de
Traditional Chinese	zh-TW
Simplified Chinese	zh-CN
Korean	ko

“Mail” on page 101 shows the changes to be made to the `ru/i18n.js` file for modifying the charset in the `i18n.js` file.

EXAMPLE 9-1 Modifying the charset in the i18n Resource File

```
i18n['client charset'] = 'iso-8859-5'
i18n['http charset'] = 'iso-8859-5'
i18n['fontface'] = 'PrimaSans BT,Verdana,sans-serif'
i18n['fontface1'] = i18n['fontface']
i18n['fontface2'] = 'Times New Roman,Times,serif'
i18n['fontface3'] = 'Courier New,Courier,mono'
i18n['nbsp'] = ' '
```

Communications Express allows you to customize text that is displayed on web pages. This text is both customizable and localizable. The system also allows you to configure text on a per domain basis.

Adding Locales

To add a locale, perform the following steps.

- Edit the `uwcdomainconfig.properties` file. This file is available in the `<uwc-deployed-path>/WEB-INF/domain directory>`
- Add a keyword for the new locale that needs to be added. For example, if you need to add a new locale, “Afrikaans”

EXAMPLE 9-2 Adding a keyword for the new locale

```
supportedLanguages=en;es;de;fr;ja;ko;zh-CN;zh-TW;Afrikaans
```

- Add a localized string for the locale name in different locales. This could be done by adding a new line in `i18n.properties` file of each locale. For example, let us add a key in `i18n.properties` file for `en` locale.

EXAMPLE 9-3 Adding a localized string for the locale name

```
uwc-common-options-preferredLanguage-Afrikaans=Afrikaans Locale
```

Changing Source Encoding

Some mail clients do not set the encoding type in MIME header before sending emails. Communications Express allows users to view a message in a different character encoding.

For example, Arabic has a number of character encoding (CP-1256, Arabic DOS, UTF8, Nafitha, Sakhr, ISO-8859, MacArabic). Communications Express allows users to change the browser encoding to be able to view the mail that they receive in different character encoding.

▼ To Enable Character Encoding

- 1 **Change directory to** `<uwc-deployed-path>/webmail/` **directory.**

- 2 **Edit** `main.js` **and uncomment the following line:**

```
//var srcchars = new Array('standard','iso-8859-6','windows-1256',  
'utf-8','big5','gb2312');
```

You can also add or remove character encoding that you want to support in the Array data type. For example:

```
var srcchars = new Array('standard','iso-8859-6','windows-1256','big5','gb2312',  
'windows-1252','iso-8859-7');
```

- 3 **Restart the web container on which Communications Express is deployed for these changes to take effect.**

Note – Users need to clear the browser cache so that the latest modifications made to the `main.js` file are picked up from the server instead of the browser's cache.

More Information Enabling Character Encoding in a Multi-tier Deployment

For multi-tier deployments, these steps should be performed in the front end directory (`<uwc-deployed-path>/webmail/`) of the Communications Express installation.

Address Book

To localize address book, the XML file that needs to be modified is the dictionary- `<locale>.xml` file that is located in the `<uwc-deployed-path>/WEB-INF/ui/html/abs` directory.

Customizing Address Book for a Specific Locale

You can customize the Address Book User Interface either for a specific locale and not under a specific domain and also for a locale under a specific domain.

▼ To Customize Address Book User Interface for a Locale and Not Under a Specific Domain

- 1 Create a directory with the locale name (For example, `ja` under `<uwc-deployed-path>/WEB-INF/ui/html/abs`)
- 2 Copy all files from `<uwc-deployed-path>/WEB-INF/ui/html/abs` to `<uwc-deployed-path>/WEB-INF/ui/html/abs/<locale>`
- 3 Make the required changes in XSL files as explained in the previous sections.

▼ To Customize Address Book for a Locale Under a Specific Domain

- 1 Create a directory with `<domain>/<locale>` (For example, `siroe.com/ja`) under `<uwc-deployed-path>/WEB-INF/ui/html`
- 2 Copy all files from `<uwc-deployed-path>/WEB-INF/ui/html/abs` to `.../abs/<locale>`
- 3 Make the required changes in `.xsl` files as explained in [Chapter 7, "Customizing General Features in Address Book."](#)

Index

Numbers and Symbols

- `<ab-images-dir>`, 21
- `<ab-ui-dir>`, 21
- `<calclient-dir>`, 21
- `<common-dir>`, 22
- `<config-dir>`, 21
- `<css-dir>`, 21
- `<domain-dir>`, 21
- `<js-dir>`, 21
- `<skin-dir>`, 21
- `<uwc-images-dir>`, 21
- 49897
 - Head2-Procedure
 - Localizing Calendar and Options, 99-101

A

- absimx, 21
- attach button, 68
- attach_fs.html file, 67
- attachments, 68

B

- browse button, 68
- browser.js, 55

C

- Cancel button, 68

- Collect button, 68
- collect_fs.html file, 67
- collecting mail from another server, 68
- comp_fs.html file, 67
- composition window, 57
- composition window functions, 58
- Customizing General Features in Address Book, 73-92
- Customizing General Features in Calendar, 35-52

D

- deployed-path, 16

F

- fldr_fs.html file, 67
- folders screen, 56

H

- Help button, 68

I

- inbox screen, 55

L

- lang/i18n.js, 63, 64, 65
- ldap_fs.html file, 67
- Linux, default base directory for, 15
- listFrameHTML(doc), 63
- localization, specific locales, 101-102
- Localizing Calendar and Options, 99-101
- Localizing Communications Express, 99-104

M

- mailbox management tab, 69
- mailui.js, 55
- main.js, 55
- mbox_fs.html file, 60, 68, 69
- message composition, 69
- message composition window, 65-66
- message display window, 62-64
- message list window, 61
- message screen, 56
- message tool bar, 64
- Messenger Express, user interface customizable features, 67-69
- Modifying the Address Book Tool Bar, 73-75
- Modifying the Mailbox Tool Bar, 73-75
- msg_fs.html file, 68, 69
- msg_fs.html file, 63, 64
- msg_fs_lr.html, 63
- msg-svr-base/html/imx, 65

O

- opts_fs.html file, 68

P

- POP server, 68
- POP user ID, 68

R

- receipt_fs.html file, 68
- receipt_fs.html file, 69
- return receipt, 69

S

- setdomain.js, 55

U

- util.js, 55
- uwc-basedir, 16
- uwc/calclient, 21
- uwc/common, 22
- uwc-common-bannerImage, 22
- uwc-common-Error-Message-icon, 22
- uwc-common-ImportExportImage, 22
- uwc-common-Info-Message-icon, 23
- uwc-common-PrintableImage, 22
- uwc-common-Question-Message-icon, 23
- uwc-common-Required-image, 22
- uwc-common-RightBrandingImage, 22
- uwc-common-SearchImage, 22
- uwc-common-Sort-Down-NonSelected-image, 22
- uwc-common-Sort-Down-Selected-image, 22
- uwc-common-Sort-Up-NonSelected-image, 22
- uwc-common-Sort-Up-Selected-image, 22
- uwc-common-Subscribed-image, 22
- uwc-common-To-Anchor-image, 22
- uwc-common-Warning-Message-icon, 23
- uwc-common-WarningImage, 22
- uwc/css, 21
- uwc/images, 21
- uwc/.js, 21

W

- WEB-INF/config, 21
- WEB-INF/domain, 21
- WEB-INF/skin, 21

WEB-INF/ui/html/abs, 21

