



Sun Java System Communications Express 6 2005Q4 Customization Guide

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

Part No: 819-2662-10
October 2005.

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

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, docs.sun.com, AnswerBook, AnswerBook2, 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 Sun™ 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.

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.

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 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux États-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 États-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.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



050926@13215



Contents

| | |
|---|-----------|
| Preface | 13 |
| 1 Introduction to Communications Express and Customization | 17 |
| Components of Sun Java Systems Communications Express | 17 |
| 2 Customizing General Features in Communications Express | 19 |
| Guidelines to be Followed Before the Customization Process | 19 |
| List of Customizable and Configurable Files for Calendar and Address Book | 20 |
| Customizing Images | 20 |
| Customizing Text | 22 |
| ▼ To Customize Text for a Domain | 22 |
| Customizing Skin | 22 |
| ▼ To Create a new Skin | 23 |
| ▼ To Modify a Skin | 23 |
| Customizing the Login Screen for English Locale | 24 |
| Replacing the Logo With a Custom Graphic | 24 |
| Example—Logo Modification | 25 |
| Changing the Title Text | 25 |
| Customizing the Banner | 26 |
| Customizing the Application Bar | 26 |
| 3 Customizing General Features in Calendar | 31 |
| Customizing Icons | 31 |
| Customizing the Order of Calendar Toolbar Icons | 32 |

| | |
|--|-----------|
| Customizing the Toolbar Images | 39 |
| Customizing Icons in Various Calendar Views | 39 |
| Customizing Events/Task Related Informative Icons | 40 |
| Customizing the Error Icon for Full Page Errors | 40 |
| Customizing Icons in Pop ups | 41 |
| Customizing Search for Events | 41 |
| Customizing Views | 42 |
| Customizing Tasks Search View | 42 |
| Customizing the Column Order in Tasks Printable View | 43 |
| Customizing Tasks Error View | 46 |
| Customizing Events view | 46 |
| Customizing Tasks View | 47 |
| Customizing Pop ups | 47 |
| Customizing Full Page Error | 48 |
| Customizing the Layout for Full Page Errors | 48 |
| | |
| 4 Customizing General Features in Mail | 51 |
| Defining Your Customization Requirements | 51 |
| Customizing the Skin to Modify Global Look and Feel | 51 |
| Customizing Text | 52 |
| Customizing Mail by Domain | 52 |
| Customizing Mail Icons | 52 |
| Adding Extra Features | 52 |
| Understanding the File Layout in Mail | 52 |
| Understanding Basic Interfaces in Mail | 54 |
| Inbox Screen | 54 |
| Message Screen | 54 |
| Folders Screen | 55 |
| Composition Window | 55 |
| Message Search Window | 56 |
| | |
| 5 Customizing User Interface Features in Mail | 59 |
| Customizing the Mailbox Tool Bar | 59 |
| Customizing the Mailbox Tool Bar | 60 |
| Example—Mailbox Tool Bar Modifications | 60 |
| Customizing Message Listing | 61 |
| Modifying Message List Window | 61 |

| | | |
|----------|---|-----------|
| | Example - Message Listing Modifications | 62 |
| | Customizing the Message Display Window | 62 |
| | Modifying Message Display Window | 62 |
| | Modify the Message Display Window to Display User Defined Header Fields | 63 |
| | Customizing the Message Tool Bar | 64 |
| | Modifying the Message Tool Bar | 64 |
| | Customizing the Message Composition Window | 64 |
| | 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 | 69 |
| | Understanding Advanced Customization | 69 |
| | List of Files to Use for Customizing the Mail User Interface | 70 |
| | Attachment Options | 70 |
| | Collect Mail from Another Server | 70 |
| | Message Composition | 71 |
| | Mailbox Management Tab | 71 |
| | Return Receipt | 72 |
| | Understanding Shared Folders | 72 |
| | ▼ To share folders | 72 |
| | Customizing the Default LDAP Attributes for Users | 73 |
| 7 | Customizing General Features in Address Book | 75 |
| | Customizing Address Book Icons and Labels | 75 |
| | Customizing Icons | 76 |
| | Customizing Style Sheets | 78 |
| | What can be Customized in Style Sheets | 78 |
| | Customizing the Main Search Page | 80 |
| | Customizing New/Edit Contact Window | 81 |
| | Customizing Phone | 82 |
| | Customizing E-mail | 85 |
| | Customizing Dates | 86 |
| | Customizing Web Addresses | 87 |
| | Customizing IM Nicknames | 88 |
| | Customizing View Contact Window | 90 |
| | Customizing New Group or Edit Group Window | 90 |

| | |
|--|------------|
| Customizing Group Details | 90 |
| Customizing the View Group Window | 91 |
| Customizing Printable Window | 92 |
| Customizing the Import/Export Address Book Window | 92 |
| 8 Customizing Communications Express for a Specific Domain | 95 |
| Customizing Default User Preferences for a Specific Domain | 95 |
| ▼ To Customize the Default User Preferences for a Domain | 95 |
| Customizing Theme for a Specific Domain and Locale | 96 |
| ▼ To Customize a Theme for a Specific Domain | 96 |
| Customizing Mail for a Specific Domain | 97 |
| Domain From URL | 98 |
| Customizing Address Book for a Specific Domain and a Locale | 98 |
| ▼ To Customize Address Book for a Specific Domain and Locale | 98 |
| 9 Localizing Communications Express | 101 |
| Localizing Calendar and Options | 101 |
| Localizing Date/Time Formats for Calendar | 102 |
| Localizing for Languages that use Characters Outside the ASCII Range | 103 |
| Localizing Mail | 103 |
| Specific Locales | 104 |
| Adding Locales | 105 |
| Localizing Address Book | 105 |
| Customizing Address Book for a Specific Locale | 105 |
| ▼ Customizing the Address Book UI for a locale and not under a specific domain | 105 |
| ▼ Customizing the Address Book for a locale under a specific domain | 106 |
| Index | 107 |

Tables

| | | |
|------------------|--|-----|
| TABLE 2-1 | List of Customizable files for Calendar and Address Book | 20 |
| TABLE 2-2 | Location of the Common images in <code>< uwc-images-dir ></code> | 21 |
| TABLE 3-1 | Customizing the Calendar Toolbar Icons | 39 |
| TABLE 3-2 | Customizing Icons | 39 |
| TABLE 3-3 | Customizing Task Related Informative Icons | 40 |
| TABLE 3-4 | Customizing Error Icon for Full Page Errors | 41 |
| TABLE 3-5 | Customizing Icons in Pop ups | 41 |
| TABLE 3-6 | Customizing Pop ups | 47 |
| TABLE 4-1 | JavaScript files that are used in Mail | 53 |
| TABLE 6-1 | Communications Express User Interface Customizable Features | 70 |
| TABLE 7-1 | Commonly used templates in search-template.xsl and search.xsl | 80 |
| TABLE 7-2 | Templates applicable for New/Edit Contact Window | 81 |
| TABLE 7-3 | Templates defined for various sections in the New/Edit group. | 90 |
| TABLE 8-1 | Directory Structure for the Domain siroe.com. | 97 |
| TABLE 8-2 | Linking Multiple Domains to few distinct brands | 98 |
| TABLE 9-1 | Localizing Date/Time Formats for Calendar | 102 |
| TABLE 9-2 | Localization Rules | 103 |
| TABLE 9-3 | Communications Express Specific Locales | 104 |

Figures

| | | |
|--------------------|--|----|
| FIGURE 2-1 | Communications Express Login Screen. | 24 |
| FIGURE 2-2 | Communications Express Logo and Link | 25 |
| FIGURE 2-3 | Example Corner Logo | 25 |
| FIGURE 2-4 | Login Screen with Customized Logo | 25 |
| FIGURE 2-5 | Communications Express Title Text | 25 |
| FIGURE 2-6 | Left Banner Image Before Customization | 26 |
| FIGURE 2-7 | Customized Left Banner Image | 26 |
| FIGURE 2-8 | Right Banner Image Before Customization | 26 |
| FIGURE 2-9 | Customized Right Banner Image | 26 |
| FIGURE 2-10 | Default application tab order | 27 |
| FIGURE 2-11 | Changing the application tab order to display Address Book, Mail, Calendar and Options | 28 |
| FIGURE 3-1 | Order of Calendar Toolbar Icons | 32 |
| FIGURE 3-2 | Changed order of the Toolbar Icons | 35 |
| FIGURE 4-1 | Communications Express Inbox Screen | 54 |
| FIGURE 4-2 | Communications Express Message Screen | 55 |
| FIGURE 4-3 | Folders Screen | 55 |
| FIGURE 4-4 | Communications Express Composition Window | 56 |
| FIGURE 4-5 | Message Search Window | 57 |
| FIGURE 5-1 | Communications Express Mailbox Tool Bar | 59 |
| FIGURE 5-2 | Communications Express Message List Window | 61 |
| FIGURE 5-3 | Communications Express Message Display Window | 62 |
| FIGURE 5-4 | Communications Express Message Tool Bar | 64 |
| FIGURE 5-5 | Communications Express Message Composition Window | 65 |
| FIGURE 7-1 | Customizing Phone | 83 |
| FIGURE 7-2 | Customizing E-mail | 85 |
| FIGURE 7-3 | Customizing the Import/Export Address Book Window. | 92 |

Examples

| | | |
|--------------------|---|----|
| EXAMPLE 2-1 | Code for customizing the Application tab order | 27 |
| EXAMPLE 2-2 | Customizing the order of the application tabs. | 28 |
| EXAMPLE 3-1 | Default code that displays the default Calendar Toolbar Icons | 32 |
| EXAMPLE 3-2 | Customizing the toolbar icon order | 35 |
| EXAMPLE 3-3 | Code before modifying the resource bundle | 42 |
| EXAMPLE 3-4 | Code After Modifying the Resource Bundle | 42 |
| EXAMPLE 3-5 | Default Ccode that Displays the Table Column Order: | 43 |
| EXAMPLE 3-6 | Changed table column ordering | 44 |
| EXAMPLE 5-1 | Altered Tool Bar Layout (mbox_fs_lr.html) | 60 |
| EXAMPLE 5-2 | Altering Tool Bar Text (en/i18n.js) | 60 |
| EXAMPLE 5-3 | Altering List Window Text (en/i18n.js) | 62 |
| EXAMPLE 5-4 | Changes made to the msg_fs_lr.html 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 | Altering Composition Window to Enable Emoticons | 66 |
| EXAMPLE 5-8 | Altering Composition Window to Disable Emoticons | 66 |
| EXAMPLE 5-9 | Aligning the User Interface from right to left | 66 |
| EXAMPLE 6-1 | Customizable LDAP attributes | 73 |
| EXAMPLE 6-2 | Display of banner advertisements depending on the user class. | 74 |
| EXAMPLE 7-1 | Example | 76 |
| EXAMPLE 7-2 | Example | 76 |
| EXAMPLE 7-3 | Customization of the label: | 77 |
| EXAMPLE 7-4 | Code before Modification of the Default Definition in Dictionary -en-xml | 77 |
| EXAMPLE 7-5 | Customized Default Definition in Dictionary -en-xml | 77 |
| EXAMPLE 7-6 | Class attribute of HTML tags | 78 |
| EXAMPLE 7-7 | Class attribute of XSL tags | 78 |

| | | |
|---------------------|--|-----|
| EXAMPLE 7-8 | Before changing the definition of a primary button display "Btnl' | 79 |
| EXAMPLE 7-9 | Class used in the XSL file for the "Close' button | 79 |
| EXAMPLE 7-10 | After changing the definition of a primary button display "Btnl' | 79 |
| EXAMPLE 7-11 | After changing class attribute reference for the desired HTML tag | 80 |
| EXAMPLE 7-12 | Code that displays Address displayed before online information | 82 |
| EXAMPLE 7-13 | Code that displays Online information before Address | 82 |
| EXAMPLE 7-14 | Default code wherein Work phone is displayed as priority 1 and home phone as priority 2 | 83 |
| EXAMPLE 7-15 | Code that displays Home as priority 1 and Work as priority 2 | 84 |
| EXAMPLE 7-16 | Default code that shows horizontally aligned fields and displays Birthday as the first field and Anniversary as the second | 86 |
| EXAMPLE 7-17 | Code that displays the field in a vertical layout and also rearranges Anniversary as the first date and Birthday as second | 86 |
| EXAMPLE 7-18 | Default code that displays SunONE as the default service | 88 |
| EXAMPLE 7-19 | Modified code that displays Yahoo as the default Service | 89 |
| EXAMPLE 7-20 | Code to be used for adding the E-mail address which is supported for a group | 91 |
| EXAMPLE 7-21 | Code before modifying the options list | 93 |
| EXAMPLE 7-22 | Code after modifying the options list | 93 |
| EXAMPLE 9-1 | Modifying the charset in the i18n Resource File | 104 |
| EXAMPLE 9-2 | Adding a keyword for the new locale | 105 |
| EXAMPLE 9-3 | Adding a localized string for the locale name | 105 |

Preface

This guide explains how to customize the look and feel of Sun Java™ System Communications Express. Although the product architecture permits an almost unlimited customization, this guide focuses on how to perform the most commonly requested 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 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

How this Book is Organized

Organization of the Sun Java System Communications Express Customization Guide

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

| Chapter | Description |
|---------------------------|--|
| This Chapter | Describes the audience, requirements, organization, document conventions, and related information. |
| Chapter 1 | This chapter provides a high-level overview on how to customize the look and feel of Sun Java Systems Communications Express. |
| Chapter 2 | This chapter explains how to customize the general features in Sun Java Systems Communications Express. |
| Chapter 3 | This chapter explains how to customize the Calendar component in Sun Java Systems Communications Express. |
| Chapter 4 | This chapter explains how to customize the Mail component in Sun Java Systems Communications Express. |
| Chapter 5 | This chapter explains how to customize the User Interface Features of the Mail component in Sun Java Systems Communications Express. |
| Chapter 6 | This chapter explains how to customize the Calendar component in Sun Java Systems Communications Express. |
| Chapter 7 | This chapter explains how to customize the Address Book component in Sun Java Systems Communications Express. |
| Chapter 8 | This chapter discusses advanced customization techniques for the mail component. |

TABLE P-1 Organization of the Sun Java System Communications Express Customization Guide (Continued)

| Chapter | Description |
|---------------------------|---|
| Chapter 9 | This chapter explains how to localize Mail, Calendar and Address Book in Sun Java Systems Communications Express. |

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.

Where to Find Related Information

In addition to this guide, Sun Java Systems Communications Express 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 2005Q4 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 Systems Communications Express

As a web-based client, the Communications Express depends on a web container for access and a browser for presentation. The Communications Express consists of three client modules - Calendar, Address Book, and Mail. The Mail, Calendar and Address Book client modules are deployed as a single application on the web container and are collectively referred to as the Unified Web Client.

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

The Address Book component of Communications Express 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.

The Mail component of Communications Express is a client side JavaScript web application. The HTML and JavaScript files are served from a separate proprietary HTTP server and are downloaded to the browser. 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 provides details on:

- [“Guidelines to be Followed Before the Customization Process” on page 19](#)
- [“Customizing Images” on page 20](#)
- [“Customizing Text” on page 22](#)
- [“Customizing Skin” on page 22](#)
- [“Customizing the Login Screen for English Locale” on page 24](#)
- [“Replacing the Logo With a Custom Graphic” on page 24](#)
- [“Customizing the Banner” on page 26](#)
- [“Customizing the Application Bar” on page 26](#)

Guidelines to be Followed Before the Customization Process

Before you start customizing the application, 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 development 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. 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.

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 here 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*

TABLE 2-1 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 | <i><config-dir></i> |
| WEB-INF/domain | All i18n.properties files. i18n files are used to internationalize the interface for multiple languages | <i><domain-dir></i> |
| WEB-INF/skin | themes.properties file | <i><skin-dir></i> |
| WEB-INF/ui/html/abs | All the .html, .xml and .xsl files | <i><ab-ui-dir></i> |
| absimx | All files such as .gif, .css, .html files | <i><ab-images-dir></i> |
| uwc/css | All stylesheets(.css files) | <i><css-dir></i> |
| uwc/images | All image files(*.gif, *.jpg) | <i><uwc-images-dir></i> |
| uwc/js | .jsp files | <i><js-dir></i> |
| uwc/calclient | .jsp files | <i><calclient-dir></i> |
| uwc/mailclient | .jsp files | <i><mailclient-dir></i> |
| uwc/common | .jsp files | <i><common-dir></i> |

Customizing Images

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

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

| Image/Icon | Property Name | Value |
|----------------------------------|---|---------------------------------------|
| Banner Image | <code>uwc-common-bannerImage</code> | <code>Masthead_s1UMC.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> |
| Warning Message Icon | <code>uwc-common-Warning-Message-icon</code> | <code>Warning_Large.gif</code> |

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

| Image/Icon | Property Name | Value |
|--------------------------|---|---------------------------------|
| 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> |

Customizing 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.

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

▼ To Customize Text for a Domain

- Steps**
1. Create `<domain-dir>/domain/example.com`
 2. Copy `<domain-dir>/domain/en/i18n.properties` to `<domain-dir>/domain/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_FR` under that domain, copy `i18n.properties` to the domain, and change the locations accordingly

Customizing Skin

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

▼ To Create a new Skin

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 do the following:

- Steps**
1. Copy `themes.properties` under `<skin-dir>` directory to `<skin-dir>/christmas.themes.properties` file that defines the various parameters of the skin
 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 set `uwc-user-attr-sunUCTheme=christmas` in the `uwcdomainconfig.properties` file.
 5. Restart the web server to apply changes.

▼ To Modify a Skin

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

- Steps**
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 parameter(s) in the `themes.properties` file.
 4. 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.
 5. Restart the web server to apply changes.

Customizing the Login Screen for English Locale

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 Communications Express 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 Corner Logo



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

Customizing the Banner

You can modify the left image and the right image in the banner. These images show 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

Customizing the Application Bar

You can modify the application bar by editing the `CalApp1BarNormal.jsp` file located at `<calclient-dir>/CalApp1BarNormal.jsp`

By default, the order in which the application tab appear, 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>
        <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",

```

EXAMPLE 2-1 Code for customizing the Application tab order (Continued)

```
"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></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' re-arrange 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 "
```

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

```
onmouseover="status='Address Book'; return true;"
onmouseout="status='';" >
<%= getLocalizedLabel(session, "uwc-calclient-toolbar-AddressBook",
    "Address Book") %></a></div>
</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>
  </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><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>
```

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

```
<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 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:

- “Customizing Icons” on page 31
- “Customizing Views” on page 42
- “Customizing Pop ups” on page 47
- “Customizing Full Page Error” on page 48

Customizing Icons

- “Customizing the Order of Calendar Toolbar Icons” on page 32
- “Customizing Icons in Various Calendar Views” on page 39
- “Customizing the Toolbar Images” on page 39
- “Customizing Events/Task Related Informative Icons” on page 40
- “Customizing the Error Icon for Full Page Errors” on page 40
- “Customizing Icons in Pop ups” on page 41
- “Customizing Search for Events” on page 41

Customizing 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.

EXAMPLE 3-1 Default code that displays the default Calendar Toolbar Icons

```
<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>
    <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" 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"
escape="false"/>',
'eventWin', 'scrollbars=yes,resizable=yes,width=700'); return false;">
<%= getLocalizedLabel(session, "uwc-calclient-toolbar-NewEvent",
"New Event") %>
</a></td>
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"
```


EXAMPLE 3-1 Default code that displays the default Calendar Toolbar Icons (Continued)

```
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>
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>
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;">" width="24" height="24"
align="absmiddle" border="0" title="<%= getLocalizedLabel(session,
```

EXAMPLE 3-1 Default code that displays the default Calendar Toolbar Icons (Continued)

```

"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>
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", "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>
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;">

```

EXAMPLE 3-1 Default code that displays the default Calendar Toolbar Icons (Continued)

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



FIGURE 3-2 Changed order of the Toolbar Icons

Example 3-2 shows the edited code that changes the order of icons in calendar toolbar to - New Task, New Event, Printable, Check Availability, Import/Export, Search for Calendar

EXAMPLE 3-2 Customizing the toolbar icon order

```
<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>
New Task
<td class="ToolbarItem" nowrap><a href="javascript:void(0)"
  accesskey="t"
onmouseover="over('newTask')" onmouseout="out('newTask')"
```

EXAMPLE 3-2 Customizing the toolbar icon order (Continued)

```

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>
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" 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" escape="false"/>',
'eventWin', 'scrollbars=yes,resizable=yes,width=700'); return false;">
<%= getLocalizedLabel(session, "uwc-calclient-toolbar-NewEvent",
"New Event") %></a></td>
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",
"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>
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>
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;">
```

EXAMPLE 3-2 Customizing the toolbar icon order (Continued)

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

Customizing the Toolbar Images

The property key names that define the location of the toolbar images are mentioned in [Table 3-1](#)

TABLE 3-1 Customizing the Calendar Toolbar Icons

| Icon | Property Name | Value |
|-------------------------|--------------------------------------|------------------------|
| New Event icon. | uwc-calclient-NewEventImage | LrlNewEvent_1_wo.gif, |
| New Task Icon | uwc-calclient-NewTaskImage | LrlNewTask_1_wo.gif |
| Check Availability Icon | uwc-calclient-CheckAvailabilityImage | LrlCheckAvail_1_wo.gif |
| Search Calendar Icon | uwc-calclient-SearchImage | LrlSearch_1_wo.gif |
| Printable Icon | uwc-calclient-PrintableImage | LrlPrintable_1_wo.gif |
| Import/Export Icon | uwc-calclient-ImportExportImage | LrlImpExp_1_wo.gif |

Customizing 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 3-2](#) shows the list of customizable icons in Communications Express.

TABLE 3-2 Customizing Icons

| Image | Property name | Value |
|---------|-----------------------------------|-------------|
| Forward | uwc-calclient-Forward-Arrow-image | forward.gif |

TABLE 3-2 Customizing Icons (Continued)

| Image | Property name | Value |
|------------------------------|--|--------------------------|
| Backward | uwc-calclient-Back-Arrow-image | backward.gif |
| Select all | uwc-calclient-Select-All-image | check_all.gif |
| Unselect all | uwc-calclient-Deselect-All-image | uncheck_all.gif |
| Sort down non selected image | uwc-common-Sort-Down-NonSelected-image | sort_down_nonsel_che.gif |
| Sort down selected image | uwc-common-Sort-Down-Selected-image | sort_down_sel.gif |
| Sort up non selected image | uwc-common-Sort-Up-NonSelected-image | sort_up_nonsel.gif |
| Sort up selected image | uwc-common-Sort-Up-Selected-image | sort_up_sel.gif |

Customizing 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 | Value |
|-----------------|-------------------------------------|-----------------|
| Notify Image | uwc-calclient-calclient-NotifyImage | LrlNotify_1.gif |
| Recurring Image | uwc-calclient-RecurringImage | LrlRecur_1.gif |
| Public Image | uwc-calclient-PublicImage | LrlPrvPub_1.gif |

Customizing the Error Icon for Full Page Errors

“[Customizing the Error Icon for Full Page Errors](#)” on [page 40](#) shows the customizable error icons for Full Page Errors

TABLE 3-4 Customizing Error Icon for Full Page Errors

| Icon | Property Name | Value |
|------------|----------------------------------|-----------------|
| Error Icon | uwc-calclient-Error-Message-icon | Error_Large.gif |

Customizing Icons in Pop ups

Table 3-5 shows the customizable icons located in `uwc/calclient/SearchCalendarData.jsp`.

TABLE 3-5 Customizing Icons in Pop ups

| Icon | Property Name | Value |
|-----------------|----------------------------------|-----------------|
| select_all | uwc-calclient-Select-All-image | check_all.gif |
| unselect_all | uwc-calclient-Deselect-All-image | uncheck_all.gif |
| subscribed icon | uwc-calclient-Subscribed-image | LrlSub_1.gif |

Customizing Search for Events

The File name including path is `<calclient-dir>/CalSearchBarPagelet.jsp`. Communications Express allows you to customize the Search for events label as shown in “Customizing Search for Events” on page 41



This is applicable for the following views:

- Day view
- Week view
- Month view
- Year view
- Events manager
- You can change the text that appears besides the search events text box
- You can also customize the overall layout

Customizing the 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 code in the resource bundle of `i18n.properties`.

[“Customizing the Text that Appears Besides the Search Events Text Box”](#) on page 41 shows the code before modifying the resource bundle

EXAMPLE 3-3 Code before modifying the resource bundle

```
uwc-calclient-toolbar-SearchForEvents=Search for Events:
```

[Example 3-4](#) shows the code after modifying the resource bundle.

EXAMPLE 3-4 Code After Modifying the Resource Bundle

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

Customizing Views

These examples show how to modify views

- [“Customizing Tasks Search View”](#) on page 42
- [“Customizing the Column Order in Tasks Printable View”](#) on page 43
- [“Customizing Tasks Error View”](#) on page 46
- [“Customizing Events view”](#) on page 46
- [“Customizing Tasks View”](#) on page 47

Customizing 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, calendar tool bar
- In the `tasksSearch.jsp` you can customize
 - The overall layout and look and feel.
 - Icons or images such as `select_all`, `unselect_all`, `sort_up`, `sort_down`, `sort_selected_up`, `sort_selected_down` gifs.
 - Task related informative icons or images such as `recurring gif`, `reminding gif`, `public task gifs`.
 - The table column ordering.
 - The edit task and view task popup window sizes and styles.

Customizing the 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 3-5 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 3-5 Default Ccode that Displays the Table Column Order:

```
<th class="TblThCl1" scope="col" nowrap><strong>
<%= getLocalizedLabel(session,
"uwc-calclient-tasks-Priority", "Priority") %></strong></th>
  <th width="35%" scope="col" nowrap><strong><%= getLocalizedLabel(session,
"uwc-calclient-tasks-Title", "Title") %></strong></th>
    <th scope="col" nowrap><strong><%= getLocalizedLabel(session,
"uwc-calclient-tasks-DueDate", "Due Date") %></strong></th>
      <th scope="col" nowrap><strong><%= getLocalizedLabel(session,
"uwc-calclient-tasks-Status", "Status") %></strong></th>
        <th scope="col" nowrap><strong><%= getLocalizedLabel(session,
"uwc-calclient-tasks-Type", "Type") %></strong></th>
      </tr>

  <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>
  <jato:content name="NonZeroTasks">
    <jato:tiledView name="TasksTiledView"
type="com.sun.uwc.calclient.TasksTileView">
      <jato:hidden name="TaskCalID"/>
      <jato:hidden name="TaskUID"/>
      <jato:hidden name="TaskRID"/>
    <tr>
      <td <jato:content name="FirstColumnStyle"/>
align="center"><jato:text name="Priority" escape="false"/></td>
      <jato:content name="IsConfidential">
      <td <jato:content name="TitleColumnStyle"/>
<%= getLocalizedLabel(session, "i18nModel",
"uwc-calclient-view-Busy", "Busy") %></td>
    </jato:content>
    <jato:content name="IsNotConfidential">
```

EXAMPLE 3-5 Default Ccode that Displays the Table Column Order: (Continued)

```
<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" 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>
</jato:content>
  <td <jato:content name="NormalColumnStyle"/>>
<jato:text name="DueDate"/>&nbsp;<jato:text name="DueTime"/></td>
  <td <jato:content name="NormalColumnStyle"/>>
<jato:text name="PercentStatusAsText" escape="false"/></td>
  <td <jato:content name="NormalColumnStyle"/>>
<jato:text name="Category" escape="false"/></td>
</tr>
</jato:tiledView>
</jato:content>
```

Example 3-6 shows the code required to change the table column ordering so that it displays in the order title, due date, status, priority and type

EXAMPLE 3-6 Changed table column ordering

```
<tr>
<th class="TblThC11" scope="col" nowrap>
```

EXAMPLE 3-6 Changed table column ordering (Continued)

```
<strong><%= getLocalizedLabel(session,
"uwc-calclient-tasks-Priority", "Title") %></strong>
</th>
<th width="35%" scope="col" nowrap><strong>
<%= getLocalizedLabel(session,
"uwc-calclient-tasks-Title", "Due Date") %></strong></th>
<th scope="col" nowrap><strong><%= getLocalizedLabel(session,
"uwc-calclient-tasks-Status", "Status") %></strong></th>
<th scope="col" nowrap><strong><%= getLocalizedLabel(session,
"uwc-calclient-tasks-DueDate", "Priority") %></strong></th>
<th scope="col" nowrap><strong><%= getLocalizedLabel(session,
"uwc-calclient-tasks-Type", "Type") %></strong></th>
</tr>

<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>
<jato:content name="NonZeroTasks">
<jato:tiledView name="TasksTiledView"
type="com.sun.uwc.calclient.TasksTileView">
<jato:hidden name="TaskCalID"/>
<jato:hidden name="TaskUID"/>
<jato:hidden name="TaskRID"/>
<tr>
<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="FirstColumnStyle"/> align="center"><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") %>"
```

EXAMPLE 3-6 Changed table column ordering (Continued)

```
alt="<%= getLocalizedLabel(session, "uwc-calclient-tasks-Recurring",
"Recurring") %>">
    </jato:content>
    <jato:content name="IsPublic">
        "
width="12" 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>
</jato:content>
    <td <jato:content name="NormalColumnStyle"/>>
<jato:text name="DueDate"/>&nbsp;  <jato:text name="DueTime"/></td>
    <td <jato:content name="NormalColumnStyle"/>>
<jato:text name="PercentStatusAsText" escape="false"/></td>
    <td <jato:content name="NormalColumnStyle"/>>
<jato:text name="Priority" escape="false"/></td>
    <td <jato:content name="NormalColumnStyle"/>>
<jato:text name="Category" escape="false"/></td>
    </tr>
</jato:tableView>
```

Customizing Tasks Error View

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

Customizing 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 image. You can change the image and then replace the changed image in the root folder.

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

Customizing Tasks View

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

Customizing 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 | <code><calclient-dir>/EditEvent.jsp</code> <code><calclient-dir>/EditEventData.jsp</code> <code><js-dir>/DeleteEventConfirmRepeatSelection.jsp</code> <code><js-dir>/SaveEventConfirmRepeatSelection.jsp</code> <code><js-dir>/EditEventButtons.jsp</code> <code><js-dir>/NewEventButtons.jsp</code> |
| View Calendar Pop up | <code><calclient-dir>/ViewCalendar.jsp</code> <code><calclient-dir>/ViewCalendarData.jsp</code> <code><js-dir>/ViewCalendarButtons.jsp</code> |

TABLE 3-6 Customizing Pop ups (Continued)

| Pop up Name | jsp files to be modified |
|---|--|
| 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> |

Customizing 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 error appears for variety of reasons. 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 the Mail component of Communications Express. This chapter has the following sections.

- “Defining Your Customization Requirements” on page 51
- “Understanding the File Layout in Mail” on page 52
- “Understanding Basic Interfaces in Mail” on page 54

Defining Your Customization Requirements

Before you customize mail for Communications Express, you need to decide your customization goals. There are several common customization scenarios.

- “Customizing the Skin to Modify Global Look and Feel” on page 51
- “Customizing Text” on page 52
- “Customizing Mail by Domain” on page 52
- “Customizing Mail Icons” on page 52
- “Adding Extra Features” on page 52

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 `xx/i18n.js`, where `xx` is the locale name for the specific language. To replace the text with a desired text, you need to look up the string from `xx/i18n.js` and modify it. The default encoding for all files is UTF-8, and you need to make sure that you use the right one, especially when the text contains non-Latin characters.

Customizing Mail by Domain

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

Customizing Mail Icons

All Image files are stored in `img/` directory. You can look up the image file name from HTML files and replace the URL with the required one.

Adding Extra Features

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

Understanding the File Layout in Mail

In mail, the static files are HTML and JavaScript files. These files have `.html` and `.js` 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 stand-alone Messenger Express and the integrated mail component of Communication Express. The JavaScript files are shared by both these interfaces, and in general provide functionalities for the application.

Files that end with `_fs_lr.html` are used exclusively in the integrated mail component of the Communication Express. This section provides a basic overview on the various files used in the integrated mail component of Communication Express.

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 shows the different JavaScript files that are used in Mail.

TABLE 4-1 JavaScript files that are used in Mail

| JavaScript file Name | Application |
|----------------------|---|
| browser.js | Detects the browser to select which .css file should be used |
| mailui.js | Maintains the common UI codes shared by messaging listing and Search message result listing |
| setdomain.js | Checks for Domain security |
| main.js | Maintains the Initialization and globally shared functions |
| util.js | Utility functions |
| add dtree.js | UI Codes for folder tree |

The frame work related files are,

- en/frame.html
- 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 (`_fs_lr.html`) to render the updated dynamic contents.

The Mail component can be perceived as a dynamic client-side HTML engine. There are not many standard HTML statements in `_fs_lr.html` files. It is the `ListFrameHTML()` function in `_fs_lr.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 `mboxFrame` to generate the dynamic HTML page.

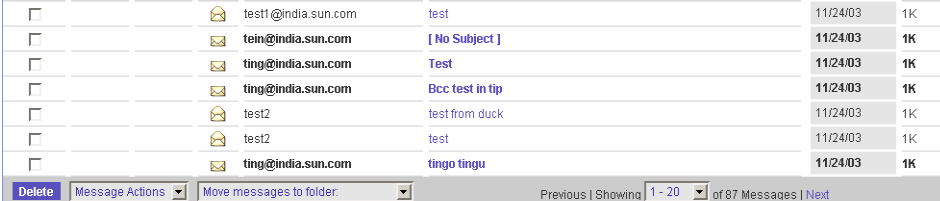
Understanding Basic Interfaces in Mail








This section provides information on the various Mail screens in Communications Express, including:

- Message Listing screen
- Message screen
- Folder Management screen
- Composition window
- Message Search window

Inbox Screen

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



| | | | | | |
|--------------------------|---|---------------------|-----------------|----------|----|
| <input type="checkbox"/> |  | test1@india.sun.com | test | 11/24/03 | 1K |
| <input type="checkbox"/> |  | tein@india.sun.com | [No Subject] | 11/24/03 | 1K |
| <input type="checkbox"/> |  | ting@india.sun.com | Test | 11/24/03 | 1K |
| <input type="checkbox"/> |  | ting@india.sun.com | Bcc test in tip | 11/24/03 | 1K |
| <input type="checkbox"/> |  | test2 | test from duck | 11/24/03 | 1K |
| <input type="checkbox"/> |  | test2 | test | 11/24/03 | 1K |
| <input type="checkbox"/> |  | ting@india.sun.com | tingo tingu | 11/24/03 | 1K |

At the bottom of the screenshot, there is a toolbar with a 'Delete' button, a 'Message Actions' dropdown menu, a 'Move messages to folder:' dropdown menu, and a status bar showing 'Previous | Showing 1 - 20 | of 87 Messages | Next'.

FIGURE 4-1 Communications Express Inbox Screen

Message Screen

The Message screen, shown in [Figure 4-2](#), displays the message selected from the Inbox screen. The Message screen 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.

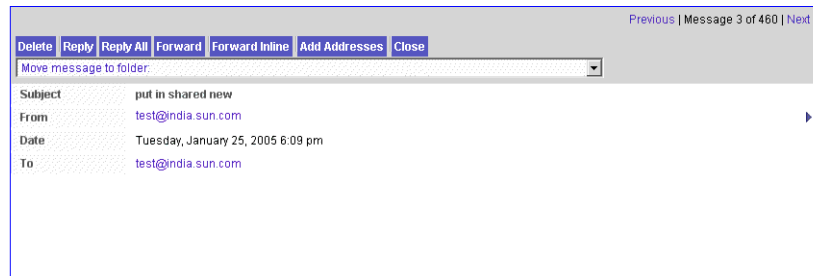


FIGURE 4-2 Communications Express Message Screen

Folders Screen

The Folders screen, shown in Figure 4-3 displays all folders that can be accessed. The Folders 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 screen, the Folders screen also enables collection of external mails.

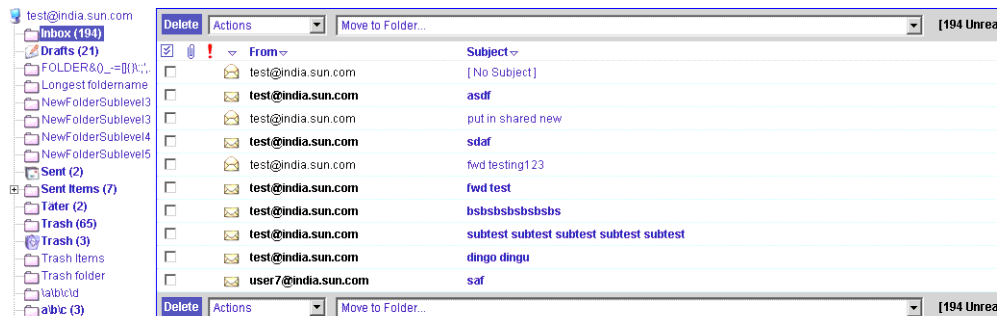


FIGURE 4-3 Folders Screen

Composition Window

The Composition window, shown in Figure 4-4, is used primarily 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, and 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 if you are using Internet Explorer and this feature is not supported in Netscape Navigator. In the Composition window you can also check the spelling. The Composition window also enables you to set the mail priority or request for a return receipt.

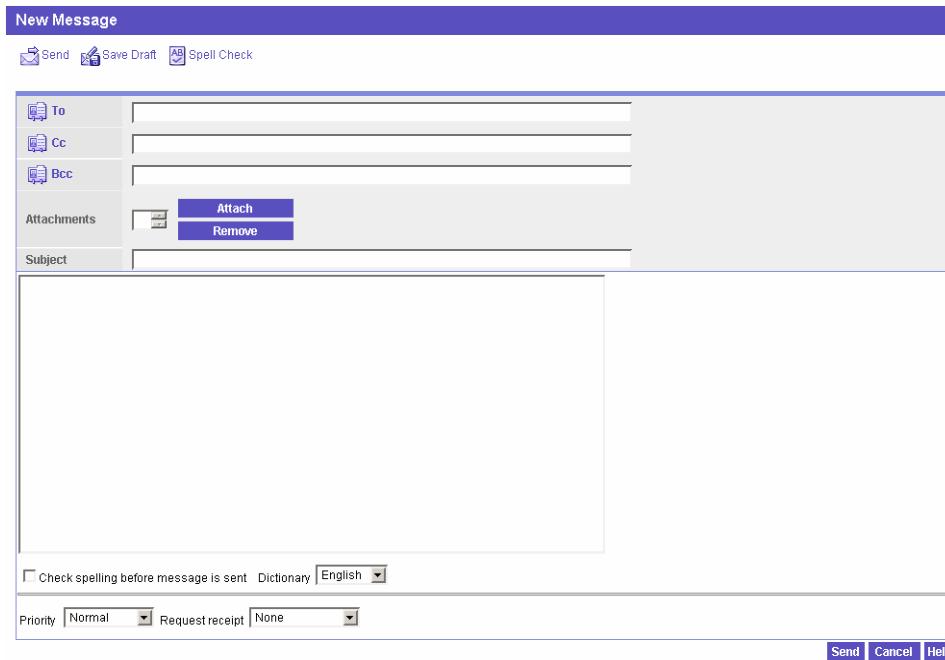


FIGURE 4-4 Communications Express Composition Window

Message Search Window

The Communications Express 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.

Search

Search for messages where

Match all of the following Match any of the following

FIGURE 4-5 Message Search Window

Customizing User Interface Features in Mail

This chapter describes how to customize user interface features of the Mail component in Communications Express.

This chapter contains the following sections:

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

Customizing the Mailbox Tool Bar

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

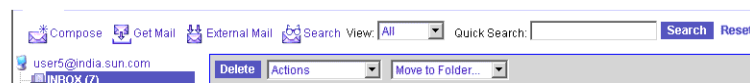


FIGURE 5–1 Communications Express Mailbox Tool Bar

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 `mbox_fs_lr.html` 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 `lang/i18n.js` file

The functions assigned by `getToolbar()` in `mbox_fs_lr.html` that handle the tool clicks are:

- 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.html` to display “Search” as the first tool and changes the text of “Get Mail” tool to “Get Messages”.

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

```
function getToolbar() {
    if (!main.loaded) return '';
    return nWMtoolbar(i18n['mbox search'], 'srch()',
        'search', 'imx/LrlSearchMsg_wo_1.gif', 'imx/LrlSearchMsg_1.gif') +
        nWMtoolbar(i18n['compose'], 'main.compose("\\new\\')',
        'compose', 'imx/LrlNewMsg_wo_1.gif', 'imx/LrlNewMsg_1.gif') +
        nWMtoolbar(i18n['get mail'], 'main.check_mail = 1;
            main.displaySpecialMbox("\\Inbox\\')',
        'getmail', 'imx/LrlGetMail_wo_1.gif', 'imx/LrlGetMail_1.gif') +
        nWMtoolbar(i18n["collect long"], 'main.collect()',
        'collect', 'imx/LrlColExtMail_wo_1.gif',
        'imx/LrlColExtMail_1.gif') +
}
```

[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'
```

Customizing Message Listing

This section describes how to modify the Message List window shown in [Figure 5-2](#).

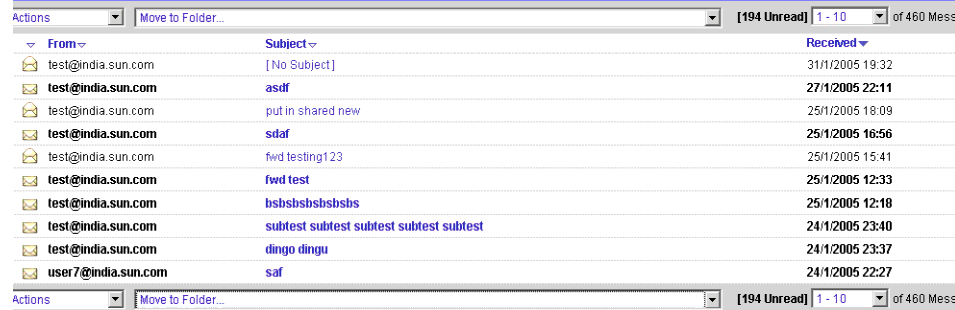


FIGURE 5-2 Communications Express Message List Window

You can modify the following in the Message List window:

- By default change the 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.html` 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 `listFrameHTML()` function links the “Collect External Mail” hyperlink to `collect()` in `main.js`. An example of such a linking is shown in - Message List Window Modifications

Example - Message Listing Modifications

To display the most recently received mails first, and 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 file `en/i18n.js`.

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](#).

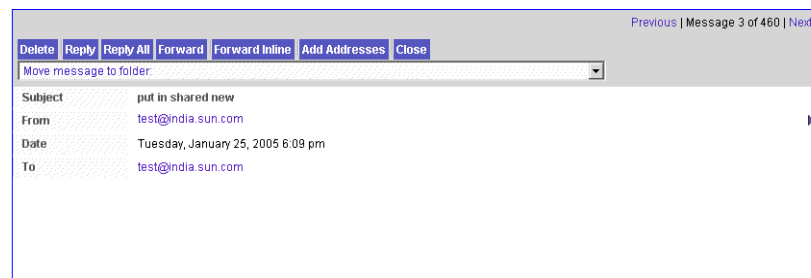


FIGURE 5-3 Communications Express Message Display Window

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.html` file.
- To customize the default text, edit the `i18n[]` values under Message Headers and Message in the `lang/i18n.js` file.

Modify 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

Edit the `listFrameHTML(doc)` function in the `msg_fs_lr.html` 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.html` file.

EXAMPLE 5-4 Changes made to the `msg_fs_lr.html` file

```
function listFrameHTML(doc) {
    ....
    var hdrstr = getHeaderStr(main.msgFrame.hdr[0], 'X-document-id')
    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 + '>' +
        extra + main.font() + hdrstr + '</td></tr>\n'
    }
    ....
}
```

Customizing the Message Tool Bar

This section describes how to customize the message tool bar



FIGURE 5-4 Communications Express 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.html` 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 file `en/i18n.js`.

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

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

Customizing the Message Composition Window

This section describes how to modify the Message Composition window

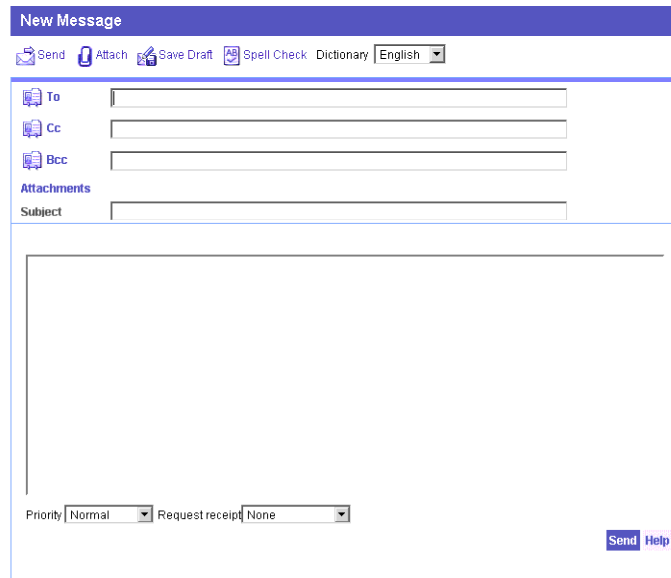


FIGURE 5-5 Communications Express Message Composition Window

You can modify the following in the Message Composition window:

- Change the location of the tools in the window
- Alter 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 appropriate files as follows:

To customize the window, edit the `compFrameHTML()` function in the `comp_fs_lr.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 `msg_svr_base/html/imx` directory.

Functionally, `compFrameHTML()` in `comp_fs_lr.html` assembles the code and assigns the functions to the graphics by calling `WMtoolbar()` in `main.js` which also handles colors and text-only versions. The `compFrameHTML()` function in `comp_fs_lr.html` generates the “To”, “Cc”, and “Bcc” control area by calling `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 '
```

The emoticons appear on the screen if the Text/HTML option is set to HTML. By default the Text/HTML option is set to Text format.

Example 5-7 shows how to edit the `main.js` file to enable emoticons.

EXAMPLE 5-7 Altering Composition Window to Enable Emoticons

```
var iconHREF = 'msg_svr_base/imx/'
```

Example 5-8 shows how to edit `main.js` file to disable emoticons.

EXAMPLE 5-8 Altering Composition Window to Disable Emoticons

```
var iconHREF = ''
```

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

- Step** ● To align the User Interface from right to left you need to remove a line from the `ar/i18n.js` file.

Example 5-9 shows the line that needs to be added to the `ar/i18n.js` file.

Example 5-9 Aligning the User Interface from right to left

```
....
```

```
i18n['dir'] = 'ltr'
```

```
...
```


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 for Communications Express user interface.

This chapter contains the following sections:

- “Understanding Advanced Customization” on page 69
- “List of Files to Use for Customizing the Mail User Interface” on page 70
- “Understanding Shared Folders” on page 72
- “Customizing the Default LDAP Attributes for Users” on page 73

Understanding Advanced Customization

In addition to the features mentioned in [Chapter 5](#), 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.

Note – This chapter does not provide code samples for advanced customizations. Also, an advanced knowledge of JavaScript and HTML is assumed.

List of Files to Use for Customizing the Mail User Interface

Table 6-1 lists the UI 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.html |
| Message listing for a selected folder | mbox_fs_lr.html |
| Message viewing | msg_fs_lr.html |
| 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.html`.

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 Tab

The mailbox management 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.html`.

HTML File Mapping for Optional Tab features

The HTML file that controls the options tab features is `opts_fs_lr.html`.

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

- Steps**
1. A user shares a folder, specifying who has permissions to the folder.
 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 on Sun Java System Communications Express.

Customizing the 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 Chapter 3 Attributes in the *Sun One Messaging and Collaboration Schema Reference Manual* at:

<http://docs.sun.com/db/prod/s1msgsrv#hic>

You might want to obtain other customized LDAP attributes from the server. For example, an 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 ONE Directory Server Deployment Guide* at:

<http://docs.sun.com/db/prod/s1dirsv#hic>

After extending the directory schema, you can add the new attribute to a User or a group entry. To add custom attribute to as User entry you need to modify the entry in `servletsconf` section using the `ldapmodify` command. For more information on `ldapmodify` command, see *Sun ONE Directory Server Configuration, Command, and File Reference* at:

<http://docs.sun.com/db/prod/s1dirsv#hic>

Depending on the type of user (that is, 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 *configutil* 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.

The following example 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 UI is handled by XML/XSL files that contain XSL tags, static HTML and JavaScripts. The XSL and JavaScript code are used to display dynamic data and perform actions or form handlings. 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 `<uwcinstall>/WEB-INF/ui/html/abs` directory. This chapter contains information on customizing the following in Address Book.

- “Customizing Address Book Icons and Labels” on page 75
- “Customizing Style Sheets” on page 78
- “Customizing the Main Search Page” on page 80
- “Customizing New/Edit Contact Window” on page 81
- “Customizing View Contact Window” on page 90
- “Customizing New Group or Edit Group Window” on page 90
- “Customizing Group Details” on page 90
- “Customizing the View Group Window” on page 91
- “Customizing Printable Window” on page 92
- “Customizing the Import/Export Address Book Window” on page 92

Customizing 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 namely,

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

All Address Book related images are located under `<uwcinstall>/absimx`. `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.

These are two examples on how you can modify the icons.

EXAMPLE 7-1 Example

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 `<uwcinstall>/absimx`.

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

EXAMPLE 7-2 Example

- To add a banner image called `AddressBook_banner.gif` you need to do the following.

- Add `AddressBook_banner.gif` file into `<uwcinstall>/absimx`
- Add a reference for the image in `commonimages.xsl` by adding the following code.

```
<xsl:variable name="AddressBook_banner.gif"
select=" ../absimx/AddressBook_banner.gif"/>
```

Then add the image in the required position in the desired `.xsl` file as

```

```

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-`<lang>.xml` file which is located under `<uwcinstall>/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 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

The following example changes the label for '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.xsl, the label is coded as following:

EXAMPLE 7-3 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 7-4 shows the code before modifying the default definition in dictionary-en.xml

EXAMPLE 7-4 Code before Modification of the Default Definition in Dictionary -en.xml

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

Example 7-5 shows the code before modifying the default definition in dictionary-en.xml

EXAMPLE 7-5 Customized Default Definition in Dictionary -en.xml

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

Customizing Style Sheets

In Address Book, style sheets play a significant role in the manner in which the UI 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 which is included using the *<link>* HTML tag in the HTML *<head>* section of XSL files. These are referenced by assigning a 'class' attribute of HTML tags.

Example 7-6 shows the class attribute using HTML tags.

EXAMPLE 7-6 Class attribute of HTML tags

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

Example 7-7 shows the class attribute using XSL tags.

EXAMPLE 7-7 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 IE and Netscape. These files exist under *<uwcinstall>/absimx* and are named as,

- *ab_css_ie5win.css* for IE
- *ab_css_ns6up.css* for Netscape

What can be Customized in Style Sheets

- You can change the definition of an existing style sheet. In this case, 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.

1. Example of changing the definitions of an existing style sheet:

The definition of a primary button display is 'Btn1' which has the following definition in the CSS file.

[Example 7-8](#) shows the code before changing the definition of a primary button display "Btn1"

EXAMPLE 7-8 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; }
```

["Displaying User Defined Header Fields" on page 63](#) shows how the class is used in the XSL file for the "Close" button.

EXAMPLE 7-9 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 7-10](#) shows the code after changing the definition of a primary button display.

EXAMPLE 7-10 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;
}
```

2. Example of adding new style sheet definitions and change the class attribute reference for the desired HTML tag in XSL files.

You may add a different class definition altogether in the CSS file. In this case, 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; }
```

In XSL file, the corresponding change would be

EXAMPLE 7-11 After changing class attribute reference for the desired HTML tag

```
<input class="EMPBtn" id="close"onblur="if (this.disabled==0)
this.className='Btnl'"onmouseover="if (this.disabled==0)
this.className='BtnlHov'"onfocus="if (this.disabled==0)
this.className='BtnlHov'"onclick="parent.close()"tabIndex"1"
onmouseout="if (this.disabled==0)
this.className='Btnl'"type="button"name="close"value="Close">
```

Customizing 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 which exist either under `search.xml` or `search-template.xml`. The `search-template.xml` file contains commonly used templates shared by the Main Search Page, Search Address Book popup and Add Addresses popup.

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

This 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 etc 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.

Customizing 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 `contact-common-editcontact.xml`

[Table 7-2](#) shows 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 | - | <code>addcontact.xml</code> and <code>editcontact.xml</code> |
| Name and Company | <code>nameAndCompany</code> | <code>lcommon-editcontact.xml</code> |
| Phone | <code>phone</code> | <code>common-editcontact.xml</code> |
| Email | <code>email</code> | <code>common-editcontact.xml</code> |
| Addresses | <code>address</code> | <code>common-editcontact.xml</code> |
| IM Nicknames | <code>IM</code> | <code>common-editcontact.xml</code> |
| Web Addresses | <code>onlineInfos</code> | <code>common-editcontact.xml</code> |
| Notes | <code>notes</code> | <code>common-editcontact.xml</code> |

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

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

[Example 7-12](#) shows the code that displays Address before Online information.

EXAMPLE 7-12 Code that displays Address displayed 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 7-13](#) shows how to display Online information before Address.

EXAMPLE 7-13 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>
```

Customizing Phone

Address Book allows you to set five types of phones which 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 their respective priority orders.

Phone

| | |
|---|--|
| Phone (1): Work <input type="text"/> | Phone (2): Home <input type="text"/> |
| Phone (3): Mobile <input type="text"/> | Phone (4): Pager <input type="text"/> |
| Phone (5): Fax <input type="text"/> | |

[Back to Top](#)

FIGURE 7-1 Customizing Phone

Example 7-14 shows the default code where Work is displayed as priority 1 and home is displayed as priority 2.

EXAMPLE 7-14 Default code wherein Work phone is displayed 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 = $priority]">
          <xsl:if test=
            "$abperson/phone[@priority =
          $priority]/@type = 'home'">

```

EXAMPLE 7-14 Default code wherein Work phone is displayed as priority 1 and home phone as priority 2 (Continued)

```

        <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 7-15 shows how to change the code so that it displays Home phone as priority 1 and Work phone as priority 2.

EXAMPLE 7-15 Code that displays Home as priority 1 and Work 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:otherwise>
      <xsl:if test="$priority = '2'">
        <xsl:attribute name="selected"/>
      </xsl:if>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>_Work</xsl:text>

```

EXAMPLE 7-15 Code that displays Home as priority 1 and Work as priority 2 (Continued)

```
</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>
```

Customizing E-mail

Address Book allows to set three types of E-mail which are defined in `phoneEmailAndIM`. The three definitions of 'email' templates are called and the priority values are passed to them. You can change their priority orders.

Email

| | | | |
|------------|-----------------------------------|------------|------------------------------------|
| Email (1): | <input type="text" value="Work"/> | Email (3): | <input type="text" value="Pager"/> |
| Email (2): | <input type="text" value="Home"/> | | |

[Back to Top](#)

FIGURE 7-2 Customizing E-mail

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

EXAMPLE 7-16 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" >
      <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 7-17 shows how to display the fields in a vertical layout and also rearranges Anniversary as the first date and Birthday as second

EXAMPLE 7-17 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">
```

EXAMPLE 7-17 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 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:with-param name="label">
      <xsl:text>_Birthday</xsl:text>
    </xsl:with-param>
    <xsl:with-param name="tabindex" select="'55'"/>
  </xsl:call-template>
```

Customizing 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 may change the order in which these addresses similar to the examples illustrated above.

Customizing 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 7-18 shows the default code that displays SunONE as the default selected service

EXAMPLE 7-18 Default code that displays SunONE as the default service

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


EXAMPLE 7-18 Default code that displays SunONE as the default service (Continued)

```
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>Yahoo</xsl:text>
</option>
```

Example 7-19 shows the modified code that displays Yahoo as the default Service

EXAMPLE 7-19 Modified code that displays Yahoo as the default Service

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

Customizing 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 etc. The file to be changed for this window is 'viewcontact.xml'

Customizing 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.

Customizing Group Details

The Group Details section displays the following fields.

- Display name
- Description

- Web URL
- Calendar URL

You may reorder the positions of these fields. Besides, you can also add the E-mail address which is supported for a group.

[Example 7-20](#) shows the code to be used for adding the E-mail address which is supported for a group.

EXAMPLE 7-20 Code to be used for adding the E-mail address which 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()"
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>

```

Customizing 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 etc. The file to be changed for this window is 'viewgroup.xml'

Customizing Printable Window

The file for this window is "abprnlist.xml". 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.

Customizing the Import/Export Address Book Window

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

Import and Export Address Book

* Indicates a required field

Import

* Import from File: Browse...

Import Format: Netscape Communicator Address Book (ldif) ▼

Import

Export

Export Format: Netscape Communicator Address Book (ldif) ▼

Export

Close Help

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
- SunONE Address Book CSV
- vCard

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

EXAMPLE 7-21 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="vcard"><xsl:text>_vCard</xsl:text></option>
  </select>
</td>
```

The options list can be changed to appear in the following order in drop down list.

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

[Example 7-22](#) shows the code after modifying the options list

EXAMPLE 7-22 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="vcard"><xsl:text>_vCard</xsl:text></option>
  </select>
</td>
```

Customizing Communications Express for a Specific Domain

Communications Express allows you to customize all the three components namely Mail, Calendar and Address Book for a specific domain. The sections contained in this chapter are,

- “Customizing Default User Preferences for a Specific Domain” on page 95
- “Customizing Theme for a Specific Domain and Locale” on page 96
- “Customizing Mail for a Specific Domain” on page 97
- “Customizing Address Book for a Specific Domain and a Locale” on page 98

Customizing Default User Preferences for a Specific Domain

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

▼ To Customize the Default User Preferences for a Domain

- Steps**
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>` directory to the newly created directory.

3. 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.
4. Restart the web server to apply changes

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

▼ To Customize a Theme for a Specific Domain

- Steps**
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" set `uwc-user-attr-sunUCTheme=christmas`. 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` to point to the customized style sheets and images.
7. For help on various parameters in this file, refer to the Appendix D of the Communications Express Administration Guide. Look for “`themes.properties`” under the “Configuration Parameters Reference” section.
8. Restart the web server to apply changes.

Customizing Mail 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 `msg_svr_base/config/html` directory.
- Populate this directory with the customized versions of the files from the default directory hierarchy.

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. The following is the directory structure for the domain `siroe.com`

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

| Directory | Description |
|--|--|
| <code>html/...</code> | // default interface |
| <code>html/imx/...</code> | // default interface |
| <code>html/en/...</code> | // default interface |
| <code>html/siroe.com/main.js</code> | // refers to <code>imx/bottle.gif</code> |
| <code>html/siroe.com/imc/bottle.gif</code> | // refers to <code>imx/bottle.gif</code> |

- After login, the server refers the user agent to pick the `main.html` file that is located in the `domain/lang` directory. The `main.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 `html/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 From 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 determines 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: `html/sesta.com/en/default.html`.

In this case a user does not have to suffix *@domain* to the username to login.

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

Customizing Address Book for a Specific Domain and a Locale

In address book, all UI related source files (xml/xsl) are located under `<uwcinstall>/WEB-INF/ui/html/abs` directory by default. To customize the UI for a domain, follow these steps.

▼ To Customize Address Book for a Specific Domain and Locale

- Steps**
1. Create a directory with a domain name, (e.g. `siroe.com`) under `<uwcinstall>/WEB-INF/ui/html`.
 2. Copy all files from `<uwcinstall>/WEB-INF/ui/html/abs` to `<uwcinstall>/WEB-INF/ui/html/<domain>`.

3. Make the required changes in XSL files as explained in the previous sections.

Although all images are located under `<uwcinstall>/absimx` by default, you may create a directory with a domain name (e.g. `<uwcinstall>/absimx/sireo.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 `<uwcinstall>/WEB-INF/ui/html/<domain>` `search-images.xsl` and `commonimages.xsl`.

Localizing Communications Express

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

- “Localizing Calendar and Options” on page 101
- “Localizing for Languages that use Characters Outside the ASCII Range” on page 103
- “Localizing Mail” on page 103
- “Adding Locales” on page 105
- “Localizing Address Book” on page 105

Localizing Calendar and Options

To localize text, you need to change the `i18n.properties` file that is located in `<domain-dir>/domain/<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>/domain`

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

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

Localizing 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

[“Localizing Date/Time Formats for Calendar” on page 102](#) contains the information on localizing Date/Time Formats for Calendar.

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 in the above table 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." [“Localizing Date/Time Formats for Calendar” on page 102](#) summarizes these rules.

TABLE 9-2 Localization Rules

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

Localizing for 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.

Localizing 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

“[Localizing Mail](#)” on page 103 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 |
| Russian | ru |
| Arabic | ar |

“[Localizing Mail](#)” on page 103 shows the changes to be made to the `ru/i18n.js` file for modifying the charset in the `i18n.resource` file.

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

```
// I18N Resource file
var i18n = new Array()
var fldr = new Array()
// DO NOT TRANSLATE AS STRINGS-JUST VALUES
i18n[\qfontface\q] = \qPrimaSans BT,Verdana,sans-serif
i18n[\qfontface1\q] = i18n[\qfontface\q]
i18n[\qfontface2\q] = \qTimes New Roman,Times,serif\q
i18n[\qfontface3\q] = \qCourier New,Courier,mono\q
i18n[\qnbsp\q] = \q&nbsp;\q
```

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-deploy-dir>/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=Afrkaans Locale
```

Localizing Address Book

To localize address book, the xml file that needs to be modified is the dictionary-`<lang>.xml` file that is located under `<uwcinstalldir>/WEB-INF/ui/html/abs` directory.

Customizing Address Book for a Specific Locale

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

▼ Customizing the Address Book UI for a locale and not under a specific domain

- Steps**
1. Create a directory with the locale name(e.g., ja under `<uwcinstall>/WEB-INF/ui/html/abs`

2. Copy all files from `<uwc-deploy>/WEB-INF/ui/html/abs` to `<uwc-deploy>/WEB-INF/ui/html/abs/<locale>`
3. Make the required changes in XSL files as explained in the previous sections.

▼ Customizing the Address Book for a locale under a specific domain

- Steps**
1. Create a directory with `<domain>/<locale>` (e.g., `siroe.com/ja`) under `<uwcinstall>/WEB-INF/ui/html`
 2. Copy all files from `<uwcinstall>/WEB-INF/ui/html/abs` to `.../abs/<locale>`
 3. Make the required changes in `.xsl` files as explained in the previous sections.

Index

Numbers and Symbols

<ab-images-dir>, 20
<ab-ui-dir>, 20
<calclient-dir>, 20
<common-dir>, 20
<config-dir>, 20
<css-dir>, 20
<domain-dir>, 20
<js-dir>, 20
<mailclient-dir>, 20
<skin-dir>, 20
<uwc-images-dir>, 20
49897
 Head2-Procedure
 Localizing Calendar and Options, 101-103

A

absimx, 20
attach button, 70
attach_fs.html file, 70
attachments, 70

B

browse button, 70
browser.js, 53

C

Cancel button, 70
Collect button, 71
collect_fs.html file, 70
collecting mail from another server, 70-71
comp_fs.html file, 70
comp_fs_lr.html, 65
composition window, 55-56
composition window functions, 56-57
Customizing General Features in Address
 Bo, 75-93
Customizing General Features in
 Calendar, 31-49

F

fldr_fs.html file, 70
folders screen, 55

H

Help button, 70

I

inbox screen, 54

J

JavaScript, 69

L

lang/i18n.js, 63, 64, 65
ldap_fs.html file, 70
listFrameHTML(doc), 63
localization, specific locales, 104
Localizing Calendar and Options, 101-103
Localizing Communications Express, 101-106

M

mailbox management tab, 71-72
mailui.js, 53
main.js, 53, 66
mbox_fs.html file, 60, 70, 72
message composition, 71
message composition window, 64-66
message display window, 62-63
message list window, 61
message screen, 54-55
message tool bar, 64
Messenger Express, user interface customizable features, 70-72
Modifying the Address Book Tool Bar, 75-77
Modifying the Mailbox Tool Bar, 75-77
msg_fs.html file, 63, 64, 70, 71
msg_fs_lr.html, 63
msg_svr_base/html/imx, 65

O

opts_fs.html file, 70, 72

P

POP server, 71
POP user ID, 71

R

receipt_fs.html file, 70, 72
return receipt, 72

S

setdomain.js, 53

U

util.js, 53
uwc/calclient, 20
uwc/common, 20
uwc-common-bannerImage, 21
uwc-common-Error-Message-icon, 21
uwc-common-ImportExportImage, 21
uwc-common-Info-Message-icon, 22
uwc-common-PrintableImage, 21
uwc-common-Question-Message-icon, 22
uwc-common-Required-image, 21
uwc-common-RightBrandingImage, 21
uwc-common-SearchImage, 21
uwc-common-Sort-Down-NonSelected-image, 21
uwc-common-Sort-Down-Selected-image, 21
uwc-common-Sort-Up-NonSelected-image, 21
uwc-common-Sort-Up-Selected-image, 21
uwc-common-Subscribed-image, 21
uwc-common-To-Anchor-image, 21
uwc-common-Warning-Message-icon, 21
uwc-common-WarningImage, 21
uwc/css, 20
uwc/images, 20
uwc/js, 20
uwc/mailclient, 20

W

WEB-INF/config, 20
WEB-INF/domain, 20
WEB-INF/skin, 20
WEB-INF/ui/html/abs, 20