



SPT Web Mail  
Application Manual Version 1.1.0

Prepared by  
Rakesh Vidyadharan

Prepared for  
Interested Developers/Deployers

Dated: January 14, 2007

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Goals</b>	<b>3</b>
<b>3</b>	<b>Features</b>	<b>3</b>
3.1	Current Features . . . . .	4
3.2	Planned Features . . . . .	4
<b>4</b>	<b>Installation</b>	<b>5</b>
<b>5</b>	<b>Customisation</b>	<b>6</b>
5.1	Localisation . . . . .	6
5.2	Dimensions . . . . .	6
5.3	Images . . . . .	7
5.4	Background Colour . . . . .	7
<b>6</b>	<b>Licences</b>	<b>7</b>
<b>7</b>	<b>FAQ's</b>	<b>7</b>
7.1	How do I copy/move/delete multiple messages? . . . . .	7
7.2	How do I increase the width of the left area in which the mail folder's are displayed? . . . . .	7
7.3	How do I change the height of the top area in which the list of messages are displayed? . . . . .	8
7.4	How do I order the message list by sender or subject? . . . . .	8
7.5	How can I get suggestions for corrections to highlighted mis-spelt words presented in the compose message window after a spell check? . . . . .	9
7.6	How can I localise spell checking? . . . . .	9
7.7	How do I add a Person to a Group in the Address Book? . . . . .	9
7.8	How do I change the default folders for deleted and sent messages that are used before a user customises their preferences? . . . . .	9
7.9	Why do you not show the message size? . . . . .	9

7.10	Why do you not show an indicator that a message contains attachments?	9
7.11	Why do you not show folder tree in menu to easily copy/move messages?	9
7.12	Why do you not use CheckBox to enable multiple selection?	10
7.13	Can I contribute to the software?	10
7.14	Why did you use Echo2/EPNG?	10
7.14.1	Why did you not use GWT in that case?	10
7.15	Why do you not use a search engine for searching messages?	11
7.16	Why do you not use a database for user data?	11
7.16.1	Why do you not use an embedded database?	11
7.17	Why do I need to maintain a independent data directory?	11
7.18	Can I modify and redistribute the application?	11
7.18.1	How can I change the SPT logo?	11
7.18.2	How can I change the logo URL?	12
7.18.3	How do I change the look-and-feel?	12
<b>8</b>	<b>Contributors</b>	<b>12</b>

## List of Tables

1	Java Requirements	5
2	Test Environment	6
3	Licence Files	8

## 1 Introduction

The SPT Web Mail Application is built primarily using the [Echo2](#) and [EchoPointNG](#) AJAX frameworks to develop the web interface. It uses [JavaMail API](#) as the middleware to connect to mail stores. The application has been tested only with [IMAP](#) stores<sup>1</sup>.

[Sans Pareil Technologies, Inc.](#) decided to develop yet another [Java](#) web mail application, since none of the freely available applications worked as we would like them to. We did not look at PHP, PERL, Python, based applications since we did not want to deal with the hassles involved in setting up all the required libraries to get those to work.

## 2 Goals

The following goals were kept in mind while developing the application:

1. The user interface should follow the Rich Internet Application concept and try and mimic the behaviour of a desktop mail client as far as possible. The interface and components were designed based upon my experience using Apple Mail.
2. Minimise the setup steps required to deploy the application.
3. Minimise dependencies on services other than a connection to the mail store and a writable filesystem.
4. Designed primarily as a single store system, while making it possible to support different mail stores on a per user basis at a later date.
5. Use only freely available and licenced libraries to implement features<sup>2</sup>.

## 3 Features

The SPT Web Mail Application application currently provides only rudimentary features that are sufficient for it to serve as an emergency mail access system. See section [3.2](#) for a list of features that will be added to the system in subsequent releases.

---

<sup>1</sup>POP stores should work fine since [JavaMail API](#) abstracts the differences between [IMAP](#) and POP.

<sup>2</sup>No library under the GPL is used by the system

### 3.1 Current Features

The following are some of the primary features available in the first release of the SPT Web Mail Application.

1. Automatically check for new messages. This feature can be turned off by the user.
2. Comprehensive mail folder management - create, rename, delete.
3. Comprehensive message management - compose, reply, reply all, forward, draft messages<sup>3</sup>.
4. Move deleted messages to standard folder used by user's preferred e-mail client.
5. Move sent messages to standard folder used by user's preferred e-mail client.
6. Address Book.
7. Search messages by subject, sender, and/or body content. Also supports search across all the mail folders. Search is implemented using [JavaMail API](#) search features and is not meant as a high performance feature.
8. Process TNEF streams and display contents/attachments in original format.
9. Multiple selection of messages for bulk copy/move/delete operations in message list view.
10. Spell checking while composing messages using [Jazzy](#) spell checking engine. See [FAQ 7.5](#).

### 3.2 Planned Features

The following are some of the primary features that will be added to the SPT Web Mail Application in future releases.

1. Support mail server connection using JNDI resource.
2. Expand properties to handle multiple server connections and enable selection of configured server set in login screen.
3. Expand preferences to include signature(s).
4. Integrate [jASEN](#) anti-SPAM engine.
5. Consider [JavaMail-Crypto](#) or [PGPMime](#) for OpenPGP and S/MIME encryption of messages.
6. Support composing and viewing e-mail as Plain text.
7. Support for user preferences such as deleting messages older than a certain date - applicable for deleted, sent and spam messages.
8. Support import (drag-and-drop if possible) of VCF and LDIF format address cards into address book.
9. Support LDAP directory look up from Address Book.

---

<sup>3</sup>Not implemented properly at present.

10. Add multiple selection for dragging and dropping **Person** entries in the **Address Book** to the **Group** if possible.
11. Allow option to reply/forward messages either as inline or attachment to new message<sup>4</sup>.
12. Experiment with custom message flags and integrate into the product.
13. Implement simple rules engine to act upon messages based on sender/subject.
14. Support public and shared namespaces that may be available to the user.
15. Experiment with keyboard shortcuts to perform common operations.

## 4 Installation

The SPT Web Mail Application is relatively easy to install. Table 1 shows the versions of Java/J2EE that are necessary for the system. Table 2 shows the environment in which the system has been tested.

Table 1: Java Requirements

Software	Version	Description
Java	1.5	Minimum version of the JRE that is required.
Servlet	1.3	Minimum version of the Servlet API that is required.
JavaMail	1.4	Minimum version of the JavaMail API that is required.

The following steps describe the process of deploying the SPT Web Mail Application to a J2EE/Servlet container. All files referred to in the instructions are available in the subversion repository.

1. Create a root directory under which persistent files used by the SPT Web Mail Application are to be stored. For example, [Sans Pareil Technologies, Inc.](#) uses a directory named `/var/data/webmail` as the root **data** directory. Ensure that this directory is writable by the system user that is used to run the container.
2. Edit the `config/properties.xml` file<sup>5</sup> and modify the IMAP and SMTP server connection properties to match your environment. Copy the modified file to the **data** directory.
3. Copy the `config/preferences.xml` file to the **data** directory.
4. Copy the `config/dictionary` directory to the **data** directory. This directory contains the dictionary files required by the **Jazzy** spell checking engine. See section 7.6 for more details.
5. Configure a JVM system property named `sptmail.data.directory` for the container. For Tomcat for example, add a line similar to the following to `$CATALINA_BASE/bin/catalina.sh`

```

JAVA_OPTS="$JAVA_OPTS -Dsptmail.data.directory=/var/data/webmail"

```
6. Make sure that you have the latest **JavaMail API** library available to the applications deployed on the J2EE/Servlet container. You can copy the `lib/mail.jar` file as appropriate if so desired.

<sup>4</sup>Currently only inline is supported.

<sup>5</sup>Standard Java 1.5 XML properties file.

7. Deploy the `mail.war` from the distribution to the container. You may have to rename the war file to something else (for example, `webmail.war`) if you wish to customise the root path at which the application is to be accessed. If you do so, edit `config/web.xml` and modify the `init-param` for `logout servlet` to match your modification.
8. Set up backup jobs for the `data directory` if desired to ensure that the server properties, address book and user preferences XML files are not lost.

Table 2: Test Environment

Software	Version	Description
Tomcat	5.5.17	Servlet container used to deploy the application.
Dovecot	1.0beta8	The IMAP server used by <a href="#">Sans Pareil Technologies, Inc.</a>
Postfix	2.3	The SMTP server used by <a href="#">Sans Pareil Technologies, Inc.</a>
Solaris	10	The operating system used on <a href="#">Sans Pareil Technologies, Inc.</a> server.
Mac OS X	10.4 (Tiger)	The client operating system used for testing.
FireFox	1.5, 2.0	The recommended browser for using the application.
Safari	2.0.4	Works mostly. Replying, forwarding messages leads to hard error in EPNG RichTextArea rendering session unusable.
Opera	9.0	Works. Some slight issues with fonts, but otherwise seems to work perfectly.

## 5 Customisation

The application may be customised to meet difference requirements to a fair degree.

### 5.1 Localisation

Most of the text displayed in the application<sup>6</sup> are configured for each UI Component in a properties file<sup>7</sup>. The values in the file may be modified to suit individual tastes. Copies of the file for other languages may be created as well which will be loaded based upon client browser language preference using regular [Java](#) property file loading rules.

### 5.2 Dimensions

The default dimensions used to build the UI components are specified for each UI Component in a properties file<sup>8</sup>. The default sizes, as well as maximum values allowed in [TextComponent](#)<sup>9</sup> are specified in this file. You can edit the values in this file to suit your preferences.

<sup>6</sup>Labels, titles, ... not including content of the message or folder.

<sup>7</sup>Located under `config/resource/localisation/Configuration.properties` in the distribution.

<sup>8</sup>Located under `config/resource/localisation/Dimensions.properties` in the distribution.

<sup>9</sup><http://www.nextapp.com/platform/echo2/echo/doc/api/public/app/nextapp/echo2/app/text/TextComponent.html>

### 5.3 Images

The images used by the application are stored under an `image` directory<sup>10</sup>. You can replace the images with your preferred images. You will probably need to update the images sizes in `com.sptci.mail.Styles` and `com.sptci.mail.style.button.*` sources.

### 5.4 Background Colour

The default red background colour that is used by application is defined in the `Background`<sup>11</sup> class. Modify as appropriate to change the colour to your preferred colour.

## 6 Licences

The SPT Web Mail Application is distributed under a [Apache 2.0](#)<sup>12</sup> licence. The third-party licences that apply to the system are included under the `config/licenses` directory in the distribution. Table 3 lists the licenses and a brief description of their use in the application.

## 7 FAQ's

Sections 7.x list some questions that may be frequently asked by people attempting to use the SPT Web Mail Application.

### 7.1 How do I copy/move/delete multiple messages?

You can select multiple messages in the message list pane. Use the appropriate keyboard modifier<sup>15</sup> button to select the messages you wish to operate on.

### 7.2 How do I increase the width of the left area in which the mail folder's are displayed?

Drag the divider to increase/decrease the width of the left area. For a more permanent solution see section 5.2.

---

<sup>10</sup>Located under `config/resource/image` in the distribution

<sup>11</sup>`src/com/sptci/mail/style/Background.java`

<sup>12</sup><http://www.apache.org/licenses/LICENSE-2.0>

<sup>15</sup>Command, Control, Alt, Shift

Table 3: Licence Files

File	Description
sptwebmail.txt	This file contains the licence under which the SPT Web Mail Application is distributed. The licence applies to all source code distributed by <a href="#">Sans Pareil Technologies, Inc.</a> as part of the system.
thinkmac.txt	This file contains the licence under which the icons used in the application are distributed.
lstnef.txt	This file contains the licence under which the <a href="#">lstnef</a> library <sup>13</sup> is distributed by the vendor.
jdom.txt	This file contains the licence under which the <a href="#">JDOM</a> <sup>14</sup> library is distributed. JDOM is used to parse/manage the Address Book files, and will likely be used in future to expand the User Preferences file.
echo2.txt	This file contains the licence under which the <a href="#">Echo2</a> library is distributed. <a href="#">Echo2</a> is the primary framework utilised to develop the User Interface.. This is the same licence under which the associated <a href="#">EchoPointNG</a> , <a href="#">Echo2FileTransfer</a> , and <a href="#">Echo2Consultas</a> libraries are distributed. Please note that some source files in the <a href="#">EchoPointNG</a> library were modified to meet the requirements of the system. The modified source files have been included as <code>config/epng.tar</code> file in the distribution.
jazzy.txt	This file contains the licence under which the <a href="#">Jazzy</a> spell checking engine is distributed.
javamail.txt	This file contains the licence under which the <a href="#">JavaMail API</a> is distributed. Please note that this library is included in the distribution only for compile time support.
activation.txt	This file contains the licence under which the <a href="#">JavaBeans Activation Framework API</a> is distributed. Please note that this library is included in the distribution only for compile time support.

### 7.3 How do I change the height of the top area in which the list of messages are displayed?

Drag the divider to increase/decrease the height of the top area. For a more permanent solution see section 5.2.

### 7.4 How do I order the message list by sender or subject?

Click on the header row of the message list to reorder the list. Click on the header again to reverse the order. Edit your preferences to specify the default ordering of messages<sup>16</sup> using `File->Edit Preferences`<sup>17</sup> menu.

<sup>16</sup>You can only order by the message date.

<sup>17</sup>The menu applies only to the application distributed by [Sans Pareil Technologies, Inc.](#). Other deployments may use different names.

### **7.5 How can I get suggestions for corrections to highlighted mis-spelt words presented in the compose message window after a spell check?**

Click the red squiggly line under the mis-spelt word to get a list of suggestions if available. Please note that suggestions will be available only to common words.

### **7.6 How can I localise spell checking?**

The spell checking dictionaries used by [Jazzy](#) are stored in the `dictionary` directory under the `data` directory. The application loads the dictionary file using usual Java localisation policy. You can add additional dictionary files to the `dictionary` directory with the appropriate name, and the system will load the appropriate dictionary to use based upon your browser preferred language.

### **7.7 How do I add a Person to a Group in the Address Book?**

Drag the icon adjacent to the Person's name to the Group you wish to add the Person to.

### **7.8 How do I change the default folders for deleted and sent messages that are used before a user customises their preferences?**

Modify the `preferences.xml` file that is stored under the application data directory and modify the values as appropriate.

### **7.9 Why do you not show the message size?**

It is on our list of things to do. It is not a critical feature, so has been ignored for the first release.

### **7.10 Why do you not show an indicator that a message contains attachments?**

It is on our list of things to do. It is not a critical feature, so has been ignored for the first release.

### **7.11 Why do you not show folder tree in menu to easily copy/move messages?**

We had implemented that, however we removed that feature due to the following reasons:

1. Building the Menu with the full folder tree takes a few seconds<sup>18</sup>. This much time to log on to the system is not considered acceptable by current standards on UI responsiveness.
2. Having many levels of nested folders leads to the entire menu not being displayable on screen.

---

<sup>18</sup>More than 5 seconds for me, since I keep my mail very well organised.

**Note:** The code for building the menu is still available in the `MailView.MenuBar`<sup>19</sup> class as `createMove` and `createCopy`.

## 7.12 Why do you not use `CheckBox` to enable multiple selection?

`CheckBox`<sup>20</sup> components will definitely meet the requirements, however it does not present the user with a Rich Internet Application experience. That said, if we are unable to<sup>21</sup> implement multiple selection in the future, we will use `CheckBox` components to satisfy the wider requirement of being able to select and operation on multiple address book entries.

## 7.13 Can I contribute to the software?

Definitely. Please see section 8 for details on how you can contribute.

## 7.14 Why did you use `Echo2/EPNG`?

`Echo2/EchoPointNG` insulates the programmer from having to deal with other rendering engines such as `XHTML 1.0`, `CSS`, `JavaScript`. Using `Echo2` helps developers avoid the impdenence mismatch between having to implement business logic in `Java` and implement display logic using a totally different technology. Additionally these API's are similar to `AWT/Swing` architecturally, reducing the learning curve for Java developers to get productive using them.

### 7.14.1 Why did you not use `GWT` in that case?

`GWT`<sup>22</sup> does provide some of the features available in `Echo22/EchoPointNG`, however `GWT` does restrict the Java classes you can use when working with it. `Echo2/EchoPointNG` on the other hand have no such restrictions, enabling you to use the entire `Java` along with any other third-party API's that help you to implement the solution you are working on.

---

<sup>19</sup>`src/com/sptci/mail/MailView.java`

<sup>20</sup>`http://www.nextapp.com/platform/echo2/echo/doc/api/public/app/nextapp/echo2/app/CheckBox.html`

<sup>21</sup>Or other contributors are unable to suggest ways to get multiple selection working.

<sup>22</sup>`http://code.google.com/webtoolkit/`

## 7.15 Why do you not use a search engine for searching messages?

This decision was made keeping in mind the fact that majority of users will use the SPT Web Mail Application only when they are unable to use their preferred E-mail client. We anticipate that users will use the web interface only periodically, thus making the effort of maintaining synchronised search indices not worth the effort<sup>23</sup>. We may expand this in future and use [Lucene](#)<sup>24</sup> or [Egothor](#)<sup>25</sup> to index messages and build background tasks that keep the search indices and the mail store synchronised.

## 7.16 Why do you not use a database for user data?

One of the goals while developing the system was to minimise dependencies on other systems. Using a database to store user preferences and address book data is easy enough to implement, but increases the dependencies for the system. The current model of using XML files meets the requirements of being easy to maintain as well as easy to persist to a database<sup>26</sup>.

### 7.16.1 Why do you not use an embedded database?

An embedded database would certainly reduce the dependency on server software requirement, however XML files are easier to backup, recover and edit manually than database files.

## 7.17 Why do I need to maintain a independent data directory?

We prefer to have all properties and other data maintained by the application in a location that is easy to backup and maintain. We could have chosen to write these files to the exploded war directory, but that is not our preferred practice. Granted, this leads to a few additional deployment steps<sup>27</sup>, however we feel the additional steps are worth the effort.

## 7.18 Can I modify and redistribute the application?

Yes, the SPT Web Mail Application is distributed under an Apache licence. You are free to modify and redistribute the application under your own brand and licence.

### 7.18.1 How can I change the SPT logo?

Replace `config/resource/image/logo.png` with your logo. See section 5 for more details.

---

<sup>23</sup>There are a number of challenges that will need to be addressed regarding keeping the search indices synchronised with the mail store if the user keeps their session very short

<sup>24</sup><http://lucene.apache.org/java/docs/index.html>

<sup>25</sup><http://www.egothor.org/>

<sup>26</sup>Most of the popular databases support storing raw XML data and querying directly using XPath/XQuery.

<sup>27</sup>This may even make it impossible for you to deploy the application at your hosting facility.

### 7.18.2 How can I change the logo URL?

Edit `src/com/sptci/echo2/SPTListener.URL` as appropriate.

### 7.18.3 How do I change the look-and-feel?

Edit the source files under `src/com/sptci/mail/style` package and its sub-packages as appropriate.

## 8 Contributors

[Sans Pareil Technologies, Inc.](#) invites people who wish to make the SPT Web Mail Application better and more feature rich to submit patches/enhancements to the product<sup>28</sup>. In particular we would appreciate contributions to the following areas:

1. [Echo2](#) experts who are familiar with writing their own custom components could help out with implementing multiple-select and bulk drag-and-drop.
2. General Java programmers can help by building importers and exporters to and from the Address Book XML file. The Address Book schema can be viewed [here](#)<sup>29</sup>. Importers from popular Desktop Address Book applications would make the system much more user friendly.
3. People with UI design skills can also contribute UI layout and design ideas<sup>30</sup>.
4. Other features that you think would make the application better.

Please send all contributions to [webmail@sptci.com](mailto:webmail@sptci.com).

---

<sup>28</sup>The only caveat is that your contributions must be under a BSD style licence.

<sup>29</sup><http://www.rakeshv.org/src/xml/schemas/addressbook.xsd>

<sup>30</sup>[Sans Pareil Technologies, Inc.](#) realises that the current SPT Web Mail Application is very raw and has not gone through any real UI design process.