

soyMAIL

Enterprise-class Web-mail for OpenVMS

Installation and Administration

Copyright

Copyright © 2005-2007 Mark G. Daniel

This program, documentation, and associated resources, come with ABSOLUTELY NO WARRANTY.

This is free software, and you are welcome to redistribute it under the conditions of the GNU GENERAL PUBLIC LICENSE, version 2

<http://www.gnu.org/licenses/gpl.html>

All trademarks within this document belong to their legitimate owners.

Author

Mark G. Daniel

Mark.Daniel@wasd.vsm.com.au

A pox on the houses of all SPAMers. Make that two poxes.

<http://wasd.vsm.com.au/>

Download

The latest release of soyMAIL is available for download from

<http://wasd.vsm.com.au/wasd/>

Publication date and software version

Published March 2007. Based on soyMAIL v1.3

OpenOffice.org

This document has been produced using OpenOffice.org 2.1

<http://openoffice.org/>

Table of Contents

1	Introduction	1
2	Installation	3
2.1	Authentication	4
2.2	Apache (SWS)	4
2.3	OSU	5
2.4	Purveyor	6
2.5	WASD	6
2.6	SMTP Relay	7
3	Configuration	8
3.1	Directives	8
3.2	[html-editor-load]	10
3.3	[logout-realm]	11
3.4	[private-access]	12
3.5	[private-request]	13
3.6	[public-access]	14
3.7	[search-control]	15
3.8	[smtp-default-host]	15
3.9	[update-last-login]	16
3.10	Conditional Configuration	16
3.11	Regular Expression Overview	18
4	Autogenous Authentication	20
4.1	Configuration	21
4.2	LOGIN.HTML	21
5	Customization	22
5.1	Locale	22
5.2	Site	22
5.3	User	23
6	soyMAIL User Files	24
7	Miscellaneous Topics	25
7.1	Run-time Problem Solving	25
7.2	Inconsistent State Data	25
7.3	Site Contact / Mailing Lists	25
7.4	General Access To Help	26

1 Introduction

soyMAIL is a native VMS application (written using the C programming language) that executes as a CGI script and allows authenticated Web access to an accounts VMS Mail. All that is required at the client end is a (relatively) modern browser supporting CSS2 (Cascading Style Sheets 2). JavaScript significantly enhances the functionality of soyMAIL but is not essential to its use. HTTP cookies are required for autogenous authentication.

soyMAIL has been built on experience gained hacking about with its progenitor yahMAIL but unlike yahMAIL was designed from the start to satisfy all the basic requirements for an *enterprise-class* Web-based email interface. It is the authors (perhaps) humble opinion that soyMAIL is a more than worthy successor as the 'son of yahMAIL'.

soyMAIL has been developed against these browsers

- Firefox 2.0 (Win32, FreeBSD, Linux)
- Mozilla 1.7 (Win32 and VMS)
- Opera 9.1 (Win32)
- MSIE 6.0 and 7.0 (Win32)

additionally reported to work with

- Camino (Macintosh)
- Safari (Macintosh)

these VMS Web server environments

- Apache (SWS) 2.1, 1.3
- OSU 3.10
- WASD 9.1,9.2

designed to be suitable for building and use on all VMS platforms

- Alpha
- Itanium
- VAX

and versions of VMS from V6.0 onwards.

soyMAIL supports

- receiving and sending
 - native VMS Mail messages
 - Internet (SMTP) mail messages
 - MIME messaging and attachments
 - HTML format message text
- message content search
- contact (address) maintenance
- VMS style mailing lists
- context-sensitive help
- user options (including supported language)

soyMAIL

soyMAIL has a *private access* mode that allows authenticated access to an underlying VMS account's email facility. This is where it provides the 'classic' web-mail functionality. It also provides a *public access* mode which requires no authentication (though it is not forbidden either) and provides controlled access to a specific folder, in a specific mail file, in a specific VMS account, intended to allow general access to a managed subset of a users Mail.

soyMAIL has been carefully and extensively optimized to perform well within the general VMS environment and when using callable Mail.

It uses VMS callable mail to access an accounts mail repository and to perform native VMS messaging.

Messages are originated via SMTP by connecting directly to an SMTP server (usually but not necessarily on the localhost), and therefore requires access to an (at least local) SMTP relay, in much the same manner as many client-based email agents.

2 Installation

soyMAIL requires some configuration before use. It is recommended that this entire document be read and carefully considered before installation and attempted usage. Please use the facilities described in section 7.1 when troubleshooting an installation.

soyMAIL is distributed as a source-code/run-time resource ZIP archive, with optional supplementary object code archives for each of the VMS platforms. For example, version one would be packaged

```
SOYMAIL100.ZIP
SOYMAIL100-AXP.ZIP
SOYMAIL100-IA64.ZIP
SOYMAIL100-VAX.ZIP
```

It may be built from the primary archive using DECC 6 and later, or link-only using in addition one of the supplementary object code archives.

Brief installation and other relevant information can be obtained from the archive using the UNZIP “-z” option.

Link-Only

A link-only build has the following alternate steps which can be used in any environment described below.

```
$ UNZIP "-V" device:[dir]SOYMAILnnn.ZIP
$ UNZIP "-V" device:[dir]SOYMAILnnn-<arch>.ZIP
$ @BUILD_SOYMAIL LINK
```

Update

An update (which does not overwrite the configuration file) has the following alternate step and can be performed in any of the environments described below.

```
$ @INSTALL UPDATE <server>
```

Run-time Components

The following table describes the run-time components. The installation procedure described below places these in the default locations the startup procedure expects to find them. Of course the default location and procedures are not mandatory. Components can be placed anywhere the site requires and a small, local startup procedure developed to use them from there.

Component	Purpose
[.OBJ_<arch>]SOYMAIL.EXE	Architecture appropriate soyMAIL script executable, to [CGI-BIN], [BIN], etc. depending on the server
[.HELP]	Help documentation, located via the logical name SOYMAIL_HELP

Component	Purpose
[.LANG]	Language files located via the logical name SOYMAIL_LANG
[.THEME]	CSS and other thematic resource files located by the logical name SOYMAIL_THEME and by the script via the path /soymail/-/theme/
SOYMAIL.CONF	Configuration file located by the logical name SOYMAIL_CONFIG.
SOYMAIL_STARTUP.COM	Startup procedure to define required logical names and install SOYMAIL.EXE image

The contents of the [.LANG], [.HELP] and [.THEME] subdirectories in the soyMAIL distribution must be available to the server and script.

The INSTALL.COM procedure should provide a default run-time installation for each of the listed server environments.

2.1 Authentication

soyMAIL can be configured to depend on the supporting Web server to perform authentication and authorization using the standard HTTP challenge/response or to use it's own (autogenous) authentication and state management.

The configuration guidelines in following sections contain both file mapping and authorization configuration examples. The file mapping configuration is always required to allow the client browser to access thematic resources. The authorization rules should not be configured when using soyMAIL autogenous login and state management. See section 4.

2.2 Apache (SWS)

This installation information is per SWS version 2.1. If you have a different version the requirements may require adjustment.

```
$ SET DEFAULT APACHE$ROOT:[000000]
$ UNZIP "-v" device:[dir]SOYMAILnnn.ZIP
$ SET DEFAULT [SOYMAIL]
$ @BUILD_SOYMAIL
$ @INSTALL INSTALL APACHE
```

After installation files are located as follows.

File	Location
SOYMAIL.EXE	APACHE\$COMMON:[CGI-BIN]
SOYMAIL_APACHE.COM	APACHE\$COMMON:[CGI-BIN]SOYMAIL.COM
SOYMAIL_STARTUP.COM	APACHE\$COMMON:[000000]
SOYMAIL.CONF	APACHE\$COMMON:[CONF]

soyMAIL

Apache requires the extra, script 'wrapper' procedure SOYMAIL.COM, so that the process-level logical name DECC\$FILE_SHARING defined in the Apache scripting environment can be deassigned before the C-RTL is activated by the soyMAIL executable. This setting interferes with some soyMAIL file operations.

Mapping and authorization examples:

```
Alias /soymail/-/ "/apache$common/soymail/"

<Location ~ "^/cgi-bin/soymail/\~">
AuthType Basic
AuthName "OpenVMS authentication"
AuthUserOpenVMS On
require valid-user
</Location>
```

Depending on the planned authorization source it may also be necessary to check that the OpenVMS authorization module is configured.

```
LoadModule auth_openvms_module modules/mod_auth_openvms.exe
```

Private access URI:

```
/cgi-bin/soymail/~
```

SWS 2.0 and 2.1 Note

There is an issue with SWS 2.0 and 2.1 when POSTing request bodies greater than 64kB. This issue has been fixed with SWS V2.1-1 released 27th September 2006.

2.3 OSU

It is suggested to place the soyMAIL kit under WWW_ROOT:[000000].

```
$ SET DEFAULT WWW_ROOT:[000000]
$ UNZIP "-v" device:[dir]SOYMAILnnn.ZIP
$ SET DEFAULT [.]SOYMAIL
$ @BUILD_SOYMAIL
$ @INSTALL INSTALL OSU
```

After installation files are located as follows.

<i>File</i>	<i>Location</i>
SOYMAIL.EXE	WWW_ROOT:[BIN]
SOYMAIL_STARTUP.COM	WWW_ROOT:[SYSTEM]
SOYMAIL.CONF	WWW_ROOT:[CONF]

Mapping and authorization examples:

```
protect /htbin/soymail/~* www_system:soymail.prot
pass /soymail/-/* /www_root/soymail/*
```

Private access URI:

```
/htbin/soymail/~
```

soyMAIL

2.4 Purveyor

Purveyor Encrypt WebServer is now explicitly excluded from soyMAIL support. A significant issue has been identified as Purveyor corrupting "text/.." response data with extraneous line-feeds if the script process SYS\$OUTPUT, 1024 character capacity mailbox, inadvertently fills during processing (the symptoms of which have been reported on a number of occasions).

Google news groups for "soymail purveyor mailbox 1024".

There is no possibility of this bug being fixed:

<http://www.process.com/techsupport/maintagreement.html>

2.5 WASD

With WASD the soyMAIL kit occupies the usual location for source under the HT_ROOT:[SRC] directory.

```
$ SET DEFAULT HT_ROOT:[SRC]
$ UNZIP "-v" device:[dir]SOYMAILnnn.ZIP
$ SET DEFAULT [.SOYMAIL]
$ @BUILD_SOYMAIL
$ @INSTALL INSTALL WASD
```

After installation files are located as follows.

<i>File</i>	<i>Location</i>
SOYMAIL.EXE	HT_ROOT:[<arch>-BIN]
SOYMAIL_STARTUP.COM	HT_ROOT:[STARTUP]
SOYMAIL.CONF	HT_ROOT:[LOCAL]

Mapping and authorization examples:

```
# HTTPD$MAP
pass /soymail/-/* /ht_root/src/soymail/*
set /cgi-bin/soymail/~* map=once

# HTTPD$AUTH
["description"]=REALM]
/cgi-bin/soymail/~* r+w
```

Private access URI:

```
/cgi-bin/soymail/~
```

2.6 SMTP Relay

soyMAIL originates SMTP Mail by directly communicating with a sites SMTP server. This requires that server to be enabled as an SMTP relay for at least the VMS system soyMAIL will be run on. The configuration for allowing relay varies on the TCP/IP and/or mail package in use (i.e. TCP/IP Services, MultiNet, TCPware, MX, PMDF, etc.)

Caution should be exercised when changing configuration to allow relay.

The last thing that you need as a result is an open relay being used for SPAM propagation!

The following example shows a possible configuration for HP TCP/IP Services (a.k.a. UCX) to provide this for the local VMS system running the soyMAIL script. The configuration option to enable relaying must be set.

```
$ TCPIP
TCPIP> SMTP SET CONFIGURATION/OPTION=RELAY
^Z
```

And an entry placed in the plain-text configuration to enable access for allowed SMTP clients (the localhost)

```
# TCPIP$SMTP_COMMON:SMTP.CONFIG
Good_clients: 127.0.0.0
```

The SMTP server that the soyMAIL script connects to is commonly running on the same (VMS) system as the script. It does not need to be. The configuration directive [SMTP-server-host] (section 3.1) can be used to specify and alternate host name or IP address for soyMAIL to connect to.

3 Configuration

soyMAIL configuration is provided using a plain-text file located using the logical name SOYMAIL_CONFIG. The configuration file must exist or all access is denied. The installation procedure copies an example configuration file allowing private access and requiring some customization. Updates do not subsequently change this file. Comments may be included by prefixing the line with a '#' character. Each directive name is delimited by '[' and ']' characters and directive parameters comprise all text until the next comment or directive opening '['. Reserved characters may be escaped using the backslash character. Leading and trailing white-space is trimmed. Comments and directives must begin in column 1.

A typical configuration file might look something like

```
#####
# private soyMAIL access
#####
[if-private]
[private-access]  */*/
[page-title]  The Company Name
[page-title2]  IT Services
[message-list-footer]
<CENTER>~ IT Services can be contacted on 82596189 ~</CENTER>
[print-header] The Company Name
[print-footer] ~ Internet, E-mail and Web Services ~
[help-about-site]
<B><A HREF="http://the.host.name/">The Company Name</A></B> provides
OpenVMS consultancy, programming and support, along with Web and Mail
hosting.
[end]
```

Comments, directive names and directive data can be seen.

3.1 Directives

The following table provides a summary of all directives, with those requiring more explanation expanded in following sections.

Directive	Description
[disk-quota-percent]	With each folder opened soyMAIL checks the users consumed disk quota. When it exceeds this percentage it adds a notification to the status information. Defaults to 85 percent . To disable completely set above 100. To display permanently set to 0.
[downtime]	Disables the use of soyMAIL and gives the specified string as a simple announcement to users connecting to soyMAIL.
[font-family-local]	Additional font families presented in the user options font selector. One per line.
[greeting]	Message presented in the status panel when a client first accesses soyMAIL, or its <i>help</i> or <i>about</i> pages. Can be used

<i>Directive</i>	<i>Description</i>
	as a welcome or warning message.
[help-about-site]	Site-specific information presented on the Help 'About' page.
[html-editor-load]	JavaScript to load and instantiate the HTML editor (see section 3.2).
[login-acme-doi]	ACME domain of interpretation (see section 4.1)
[login-autocomplete]	browser autocompletion of login credentials (see section 4.1)
[login-idle]	seconds idle before credential reentry (see section 4.1)
[login-language]	language for login page (see section 4.1)
[login-page]	file-system location for login page (see section 4.1)
[login-secret]	encryption key for credential cookie (see section 4.1)
[login-verify]	seconds between credential verification (see section 4.1)
[logout-realm]	Enables the logout button and functionality (see section 3.3).
[message-list-footer]	Site-specific information (HTML text) presented in a separate panel below the folder message listing.
[page-title]	Superior line in main menu panel. Titles all pages.
[page-title2]	The text in [page-title] is forced left and the [page-title2] text is forced right.
[print-header]	Text header for printed mail messages.
[print-footer]	Text footer for printed mail messages.
[private-access]	Allows mapping between authenticated user (CGI remote-user) and a VMS username (see section 3.4).
[private-request]	Indicates this request is for private access (see section 3.5).
[public-access]	Permits and maps request path strings to VMS Mail user names, mail file and folders (see section 3.6).
[public-options-default]	Allows the soyMAIL administrator to set public access 'personal option' settings(required to escape each leading '[' with a '\').
[public-request]	Indicates this request is for public access. Complements the [private-request] directive.
[search-control]	Controls designed to limit the impact of intensive searching activity on system resources (see section 3.7).
[smtp-default-host]	Specifies a host or domain name and makes an unqualified user address such as Mark.Daniel into an RFC (Internet-style) address such as Mark.Daniel@vsm.com.au (see section 3.8).
[smtp-from-host]	Used when constructing the 'user@from.host.name'.
[smtp-server-host]	Host name/address of SMTP server soyMAIL connects to for Internet mail transport (defaults to localhost).

Directive	Description
[soymail-at-title]	Site description provided in title bar of browser window (defaults to "soyMAIL @ the.server.name").
[update-last-login]	Update SYSUAF last-login with each initial access (see section 3.9).
[user-options-default]	Allows the soyMAIL administrator to preset some options (e.g. language) by providing options configuration text against this directive(it is required to escape each leading '[' with a '\').
[user-options-override]	Allows the soyMAIL administrator to override user selected options by providing options configuration text against this directive (it is required to escape each leading '[' with a '\').
[vms-occluded]	'Hides' obvious VMS-specific aspects of soyMAIL (e.g. VMS options panel, the extract button on the message read page).

3.2 [html-editor-load]

soyMAIL supports plug-in JavaScript HTML editors for HTML message composition.

So far it has been tried successfully with

FCKeditor v2.2 <http://www.fckeditor.net/>
<http://sourceforge.net/projects/fckeditor/>

TinyMCE v2.0 <http://tinymce.moxiecode.com/>
<http://www.moxiecode.com/>

Make your choice. Mine is TinyMCE. Install it independently of soyMAIL. Check configuration requirements for the editor. Modify the soyMAIL configuration directives to provide the path to the editor and the initialization JavaScript code, both of which will be provided in the editor installation guidelines, and voila! The function *htmlEditorLoad()* must be present (even if empty as in the case of TinyMCE), being an *onload=* target for the document.

TinyMCE Example

```
[html-editor-load]
<script type="text/javascript"
      src="/tinymce/jscripts/tiny_mce/tiny_mce.js">
</script>
<script type="text/javascript">

tinyMCE.init({
  mode : "exact",
  elements : "compose_text",
  theme : "advanced",
  plugins: "preview,print,insertdatetime",
```

soyMAIL

```
theme_advanced_buttons1_add: "fontselect,fontselect",
theme_advanced_buttons2_add: "preview,print",
theme_advanced_buttons3_add: "insertdate,inserttime",
theme_advanced_buttons3_add_before: "forecolor,backcolor"
});

function htmlEditorLoad() { /* nothing required for TinyMCE! */ }
</script>
```

This would also require a server mapping rule into the TinyMCE installation. For the above configuration and WASD it might be something like

```
pass /tinymce/* /$1$disk/tinymce/*
```

FCKeditor Example

```
[html-editor-load]
<script type="text/javascript" src="/FCKeditor/fckeditor.js"></script>
<script type="text/javascript">
function htmlEditorLoad()
{
    var oFCKeditor = new FCKeditor( 'compose_text' ) ;
    oFCKeditor.BasePath = "/fckeditor/" ;
    oFCKeditor.ReplaceTextarea() ;
}
</script>
```

Along with a similar mapping requirement

```
pass /fckeditor/* /$1$disk/fckeditor/*
```

3.3 [logout-realm]

This configuration directive only applies when using Web server authorization. It is not needed and is not used when using soyMAIL own authentication and state management (section 4). The following description only applies to Web server HTTP authorization.

The soyMAIL [logout] button and functionality is based on a Kludge that tries to hoodwink the browser into thinking its cached credentials are no longer valid. It does this by returning an HTTP 401 response (authentication required) itself as a response to hitting the [logout] button. The idea is to present to the browser a refusal to use the supplied username/password against the request path whereupon the browser purges the entry from its credential cache.

As described above this is a Kludge with a capital K. Why HTTP/1.1 didn't include a 418 (authorization canceled) – or some mechanism such as this – I'll never know! Not only are there inconsistencies in the way browsers handle this (hence the credential clear, [ok] and final [cancel]) there are some issues in getting a CGI application issued *Status: 401 authorization required* through the server sufficiently unscathed and functional as to be recognised by the browser as an authorization refusal. So...

WASD handles all this with its usual panache :-)

VMS Apache and OSU need some additional working-around. Hence the [logout-realm] directive. Unless this is set the [logout] button is greyed-out (italicised) and the functionality disabled. This must be set to exactly the same string used by the

soyMAIL

authorization realm configured against the soyMAIL path in the servers configuration. If it not exactly the same string some servers go into infinite loops, some browsers re-request ad-infinitum, etc. You have been warned!

For an Apache configuration of

```
<Location ~ "/cgi-bin/soymail/~">
AuthType Basic
AuthName "OpenVMS authentication"
AuthUserOpenVMS On
require valid-user
</Location>
```

the soyMAIL configuration must be set

```
[logout-realm] OpenVMS authentication
```

For an OSU configuration of

```
protect /htbin/soymail/~* www_system:soymail.prot
```

and a soymail.prot configuration of

```
<realm> VMS account
*@*.*.*. * *
```

the soyMAIL configuration would be set

```
[logout-realm] VMS account
```

The soyMAIL configuration directive may also be set to a single hyphen to disable the logout button and functionality.

3.4 [private-access]

Private-access is a term used to describe a client making authenticated access to an underlying VMS accounts email facility.

The private access URI must have been made subject to either Web server authorization, or to soyMAIL autogenous authentication (section 4). If there is no browser username/password dialog, or no login page, then it's not configured correctly!

soyMAIL identifies private access when the path has the leading characters "/~". For example, in the case of Apache and WASD

```
/cgi-bin/soymail/~
```

Alternatively, a site-specific private sentinel can be configured using the [private-request] configuration directive (section 3.5).

There needs to be a explicit configuration directive to enable private access and optionally to map between authenticated users and VMS usernames. The general format is

```
<remote-user>/<realm>/<vms-username>
```

The *realm* allows WASD authentication realms to be factored into the mappings but almost always will remain an asterisk.

If there is a one-to-one correspondence between the Web-server authenticated user name

soyMAIL

(as it is common to use some form of SYSUAF-based authentication this is usually the case) then a simple rule against the directive is enough to allow any user access to Mail.

```
[private-access]  */**/*
```

Specific accounts can be denied access by deliberately disrupting the mapping. In this case the SYSTEM and ANOTHER accounts are mapped to non-existing accounts `_SYSTEM` and `_ANOTHER`.

```
[private-access]
system/*/_system
another/*/_another
*/**/*
```

Using the same mechanism a non-VMS-account remote user may be mapped into the mail of an existing VMS username.

```
[private-access]
freddo/*/_nurkf
jd/*/_doej
*/**/*
```

Postmaster

POSTMASTER is a flag that can be placed against any user name. It allows a username to be specified in the soyMAIL path different to that of the authenticated username (normally this will result in a soyMAIL “insufficient privilege or object protection violation” fatal error). For example

```
/cgi-bin/soymail/~daniel/
```

This postmaster can then do anything using soyMAIL the specified username can do. Such an account is flagged in the following manner.

```
[private-access]
whomever/*/(POSTMASTER)
*/**/*
```

Note that POSTMASTER here is not an account. It is a soyMAIL characteristic flagged against the username. The parentheses are required.

3.5 [private-request]

By default the leading characters “/~” of the path indicates to soyMAIL a private access request. This directive overrides that tilde sentinel and allows any request to be recognized for private access. It intended to be used inside a conditional configuration test (section 3.10) based on some characteristic of the request reflected in a CGI variable.

The following example shows a configuration test for the presence of a `REMOTE_USER` variable indicating it is a request subject to server authorization.

```
[if-CGI-remote_user]
[private-request]
[end-if]
```

This effectively directs soyMAIL to consider any such authorized request is private access.

soyMAIL

The second example shows a test based on the request script name and assumes that the server has mapped the path /mail onto the soyMAIL executable.

```
[if-CGI-script_name]  ^^/mail$  
[private-request]  
[end-if]
```

The first leading caret indicates it is a regex pattern; the second a regex 'beginning of line' symbol; then the string used to activate soyMAIL; and a regex 'end of line' dollar symbol. This makes it an exact match test.

Note that the [private-request] directive must be specified before the use of any directives that rely on recognition of private or public access (e.g. [if-private], [if-public], [private-access], [public-access], etc).

3.6 [public-access]

This facility is intended to allow general access to a managed subset of a users Mail. Public access requires no authentication (though it is not forbidden either). There are three variations on public access. The first rigidly controls access to a specific folder, in a specific mail file, in a specific VMS account. The second, a wildcard folder specification, allows access to any folder in the specified mail file and account. The third, also a wildcard specification, prohibits access to the account MAIL and NEWMAIL folders.

The general format is

```
/<public-path>/ /<username>/<mail-file>/<folder-name>/
```

with the full wildcard variant

```
/<public-path>/ /<username>/<mail-file>/*/
```

and the prohibited MAIL and NEWMAIL wildcard variant

```
/<public-path>/ /<username>/<mail-file>/!*/
```

Both wildcard variants allow an initial folder to be specified

```
/<public-path>/ /<username>/<mail-file>/*<folder>/  
/<public-path>/ /<username>/<mail-file>/!*<folder>/
```

The public path string is used in the URL and need not be related to any part of the real components of the mail access.

```
[public-access]  
/an_example/ /NURKF/MAIL/PUBLIC/  
/another_example/ /DOEJ/ARCHIVE/MAIL/  
/public/ /LARRY/MAIL/!*MY_PUBLIC/
```

The first rule would map the URL

```
http://the.host.name/cgi-bin/soymail/an_example/
```

to VMS account NURKF, the default mail file MAIL.MAI and the PUBLIC folder in that.

soyMAIL

The second rule maps the URL

```
http://the.host.name/cgi-bin/soymail/another_example/
```

to the JOED account, mail file ARCHIVE.MAI and folder MAIL in it.

The third rule maps the URL

```
http://the.host.name/cgi-bin/soymail/public/
```

to the LARRY account, mail file MAIL.MAI and an initial folder MY_PUBLIC, with access allowed to all other folders except MAIL and NEWMAIL.

3.7 [search-control]

Mail message searching can be a very compute and I/O intensive activity. Essentially soyMAIL searching is performed on two levels with significantly different expenses.

- Message header content. From, to, cc, subject and date parameters are all available via callable Mail without needing to access the message body. This makes searching on these very inexpensive.
- Message body content. This requires not only reading the body of the message but also commonly decoding the MIME content of messages and can be quite expensive. In addition performing a textual search of the body of a message is obviously a source of intense CPU activity.

Regular expression matching is significantly more CPU intensive than keyword matching.

The following parameters to the [search-control] directive allow fine control of the extent of search capability to assist in managing the system impact of this activity. Conditional configuration (see below) makes it possible to apply these to some requests and not others. One or more, space-separated, can be used against the directive.

Parameter	Description
full	Is the default if [search-control] is not configured.
header-only	As described above this is quite lightweight.
keyword-only	Disable regular expression matching. It is still available to configuration directives.
priority=<integer>	Once message body searching begins this moves the script process priority to that specified (0..4) and restores it when searching completes.
none	Disables searching completely (and is obviously the least expensive :-)

3.8 [smtp-default-host]

This directive allows a host/domain name to be automatically appended to an unqualified user name (i.e. an address with a local but no @domain part). With this set to *the.host.name* entering an address of *Mark.Daniel* would result in a send to *Mark.Daniel@the.host.name*.

soyMAIL

soyMAIL adds the default host/domain part before sending or whenever the page is refreshed. The modification to the address can be requested at anytime by selecting the [compose] button and thereby refreshing the page.

Of course setting this directive disables a default send via VMS Mail. When set VMS Mail can still be used by prepending a node name to the address (e.g. '0::DANIEL', 'DELTA::DANIEL', etc.)

3.9 [update-last-login]

For sites with a requirement to track continued account usage this directive results in the SYSUAF interactive or non-interactive (default) last-login datum being updated with each initial access. An initial access is defined as a startup GET request from entering the private access URL into the browser or selection of a bookmark/favourite. Continued use of an established session (using the buttons or new mail check facility) does not result in updates to the last-login date/time.

Parameter to this directive should be INTERACTIVE or NON-INTERACTIVE.

3.10 Conditional Configuration

soyMAIL has a useful facility allowing dynamic configuration of soyMAIL processing based on request characteristics and CGI variable content. This allows a single configuration file to support multiple site appearances or capabilities.

Conditional directives begin with a test. Some are booleans. For CGI tests it either looks for the keyword provided with the test directive in the specified CGI variable value, or uses it as a regular expression (regex) to match against the variable value (EGREP compliant). A regular expression must be prefixed by a caret character (^) which of course is not used as part of the expression. If a CGI variable doesn't exist it is treated as an empty string.

<i>Directive</i>	<i>Description</i>
[if-CGI-<name>]	If the CGI variable specified by <name> matches the directive string then process the directives to the corresponding [if-end].
[if-not-CGI-<name>]	If the CGI variable specified by <name> <u>does not</u> match the directive string then process the directives to the corresponding [if-end].
[if-private]	If soyMAIL is being used to access private mail.
[if-public]	If being used to access public mail.
[if-end]	Terminates a block started by a matching [if-...] directive.
[end]	WARNING: Terminates all further configuration processing at that point. This is intended merely to save a few CPU cycles. Deploy inside relevant [if-...] directives.

Keyword Example

```
[if-CGI-http_host] the.host.name
[if-CGI-server_name] the.server.name
[if-not-CGI-remote_user] R_J_ECUREUIL
```

Equivalent Regex Example

```
[if-CGI-http_host]    ^.*the\.host\.name.*  
[if-CGI-server_name]  ^.*the\.server\.name.*  
[if-not-CGI-remote_user] ^.*R_J_ECUREUIL.*
```

Regex Equivalents of Common Wildcards

<i>asterisk ('*')</i>	<i>matches zero or more characters</i>	<i>period then asterisk ('.*')</i>
<i>question mark ('?')</i>	<i>matches any single character</i>	<i>period ('.')</i>
<i>Percentage ('%')</i>	<i>matches any single character</i>	<i>period ('.')</i>

'Boolean' Tests

Empty and non-empty strings may be tested for using an empty parameter to the directive.

If the variable contains a value then the following test will be true. If the variable does not exist or has an empty value then it will be false.

```
[if-CGI-<variable_name>]
```

If the variable does not exist or contains an empty value then this second test will be true. If it contains a value then it will be false.

```
[if-not-CGI-<variable_name>]
```

Conditional Configuration Example

```
# first directives for private access  
[if-private]  
[page-title] The Page Title  
[if-CGI-server_name] www.site1.com  
[message-list-footer] For more information on Site 1 see ...  
[if-end]  
[if-CGI-server_name] www.site2.com  
[message-list-footer] For more information on Site 2 see ...  
[if-end]  
[if-end]  
#  
# then directives for public access  
[if-public]  
[page-title] The Public Page  
[message-list-footer] Just an public example!  
[public-access] /public-2005/ /WEBMASTER/MAIL/PUB2005/  
[if-end]
```

3.11 Regular Expression Overview

soyMAIL employs the GNU RX1.5 regular expression package. Evaluation is done using case-insensitive, EGREP-compatible matching. A detailed tutorial on regular expression capabilities and usage is well beyond the scope of this document. This summary is only to serve as a quick mnemonic.

Also see http://en.wikipedia.org/wiki/regular_expression

soyMAIL regular expressions support the following set of operators.

<i>Description</i>	<i>Usage</i>
Match-self Operator	Ordinary characters
Match-any-character Operator	.
Concatenation Operator	Juxtaposition
Repetition Operators	* + ? {}
Alternation Operator	
List Operators	[...] [^...]
Grouping Operators	(...)
Back-reference Operator	\digit
Anchoring Operators	^ \$
Backslash Operator	Escape meta-character (e.g. \ ^ . \$ [()

The following operators are used to match one, or in conjunction with the repetition operators more, characters of the target string. These single and leading characters are reserved meta-characters and must be escaped using a leading backslash ("\") if required as a literal character in the matching pattern.

<i>Expression</i>	<i>Purpose</i>
^	Match the beginning of the line
.	Match any character
\$	Match the end of the line
	Alternation (or)
[abc]	Match only a, b or c
[^abc]	Match anything except a, b and c
[a-z0-9]	Match any character in the range a to z or 0 to 9

Repetition operators control the extent, or number, of whatever the matching operators match. These are also reserved meta-characters and must be escaped using a leading backslash if required as a literal character.

soyMAIL

<i>Expression</i>	<i>Function</i>
*	Match 0 or more times
+	Match 1 or more times
?	Match 1 or zero times
{n}	Match exactly n times
{n,}	Match at least n times
{n,m}	Match at least n but not more than m times

Checking Regular Expressions

soyMAIL can be used at the command line to check the results of regular expression pattern matching. This can assist with creating conditional configuration. Assign soyMAIL as a foreign verb and use as illustrated.

```
$ SOYMAIL /REGEX "<text to be matched>" "<pattern>"
```

4 Autogenous Authentication

Access to private Mail via soyMAIL always requires user authentication. This may be performed as HTTP authorization by the supporting Web server (and passed as CGI variable REMOTE_USER), or by having soyMAIL maintain it's own authentication state. This section describes the latter.

The main advantage in allowing soyMAIL to manage it's own authentication environment is a clearly defined login-usage-logout life-cycle, something at best a kludge with HTTP authorization. It provides a one-click [logout] button that really works! :-)

The client browser must support and have enabled HTTP cookie functionality. A cookie is used to store and propagate the authentication state. The cookie expires at the close of the browser. The cookie value is encrypted using the industry standard RC4 algorithm. This data is only present as plain values inside the processing of soyMAIL.

soyMAIL authentication

- accepts the VMS username and password via it's own login page
- verifies those credentials
- maintains verified credentials for the duration of the session
- session is terminated by using the [logout] button, or fully closing the browser

In addition soyMAIL

- continues to verify the credentials periodically
- will request the credentials be reentered after an idle period

The authentication mechanism depends on the VMS platform and version.

- For Alpha and Itanium with VMS V7.3 or later it uses the ACME authenticator and \$ACM system services. This is a vendor maintained service and so may be considered as-good-as bullet-proof. It supports everything ACME supports including DOI, external authentication, intrusion management, etc.
- For VAX and VMS versions earlier than V7.3 it uses a combination of \$SCAN_INTRUSION, \$GETUAI and \$HASH_PASSWORD to parallel most of the relevant functionality provided by ACME. These include intrusion management and account login constraints. It's potentially not as bullet-proof as the vendor solution.

With autogenous authentication any account that can normally login to the supporting VMS system can access soyMAIL. To restrict an account it is necessary to deny access to soyMAIL using the [private-access] directive (section 3.4).

Technical detail on the encryption algorithm, authentication code, cookie structure, etc., can be obtained from the descriptive prologue of the LOGIN.C source code module.

4.1 Configuration

If Web server authorization for soyMAIL is currently configured then disable this before proceeding further. Behaviour with both enabled is indeterminate.

Autogenous authorization is enabled by configuring a value against the soyMAIL configuration directive [login-secret]. This is used as a key to encrypt the soyMAIL credential cookie value. When this directive has a value against it soyMAIL attempts to establish it's own authentication state. As this is an encryption key care should be taken to ensure it is

- as long as practicable (soyMAIL insists on a minimum of 24 characters)
- comprises a 'random' mix of mixed-case alphanumeric and punctuation characters
- kept secure (SOYMAIL.CONF is not world-accessible, etc.)

An example configuration might be

```
[login-secret] aic84+-[QsfOPa,dk;'12ndjve64_90mhd43m*2$
```

The rest of the configuration parameters have defaults described in the following table.

Directive	Description	Default
[login-acme-doi]	ACME Domain Of Interpretation	VMS
[login-autocomplete]	enable (allows) browser autocompletion of login page username/password details	disable
[login-idle]	session idle period before user is required to re-enter their credentials	3600 seconds (1 hour)
[login-language]	language to use on login page	en
[login-page]	file-system location of login page	SOYMAIL_THEME:LOGIN.HTML
[login-secret]	encryption key for cookie (see above)	<i>none</i>
[login-verify]	period between re-verification of current login credentials	300 seconds (5 minutes)

4.2 LOGIN.HTML

The HTML used in the login page is derived from a file. The default is [.THEME]LOGIN.HTML and contains a functional login form and layout. This can be customized to include a site theme, logo, information, etc. Care must be taken not to disturb any of the core functionality of the HTML form, form fields, etc.

So that subsequent updates to not overwrite modifications to the base LOGIN.HTML It is suggested to copy this to something like _LOGIN.HTML and then set that as the login page via soyMAIL configuration.

```
[login-page] SOYMAIL_THEME:_LOGIN.HTML
```

5 Customization

soyMAIL provides three levels of customization.

- Locale
- Site
- User

5.1 *Locale*

The `[.LANG]` directory contains the default English language message file, `EN.TXT`. This can be copied to another, language-indicative file name and the text of the messages modified appropriately. The content of these files is not intended to be used for site-local changes to messages. Directives in the soyMAIL configuration file are for this purpose. The language files are global components of the soyMAIL distribution.

If you wish to put a language-specific message file together for soyMAIL please contribute it back into the soyMAIL community. Note that messages can be used in all sorts of contexts, particularly inside string literal quotes - both single and double. It is therefore necessary to substitute the HTML entities `"`, `’`, etc., for anything that might be misinterpreted as JavaScript code quotes (i.e. `"` (0x22) and `'` (0x27)). If soyMAIL reports a message file fatal error the `SOYMAIL$WATCH` facility (section 7.1) can be used to help determine the underlying problem.

The `[.HELP]` directory contains the default English language, on-line, content-sensitive help files. These contain the help information and HTML markup. Each files name contains an indication of the language, where the English version might be `HELP_EN.HTML`, the French language version `HELP_FR.HTML`, etc. These files are dynamically accessed and composed by soyMAIL when a user accesses on-line help. The users language option is used to search for a possible file with that language indicated in the name. If not found it supplies the default English language version. One or (preferably) all of each of the help topic files can be copied to a language-specific instance and translated. As with message files please make any such resources available to the general soyMAIL community.

5.2 *Site*

The soyMAIL configuration file has several directives intended to allow site-specific information to be included in soyMAIL pages at appropriate locations.

- `[help-about-site]`
- `[help-local-text]`
- `[message-list-footer]`
- `[page-title]`
- `[page-title2]`
- `[print-header]`
- `[print-footer]`

soyMAIL

- [soyMAIL-at-title]

The SMTP host and apparent source of messages can be specified.

- [SMTP-server-host]
- [SMTP-from-host]

The obviously VMS-specific portions of soyMAIL (e.g. the VMS options panel, the extract button) may be 'hidden'.

- [VMS-occluded]

Local User Option Defaults

Users options may be defaulted and overridden.

- [user-options-default]
- [user-options-override]

For example; to provide a language default other than English, perhaps German

```
[user-options-default]
\[language] de
```

To force a site to use a particular (perhaps corporate) theme

```
[user-options-override]
\[theme] _draco_corp
```

Two notes:

- The user-option directives used as parameters in these examples must be escaped.
- Themes beginning with an underscore will never be overwritten by a soyMAIL update. soyMAIL never will include themes beginning with an underscore.

Local Help

In addition a site-specific help file can be created in the [.HELP] directory. It must be named `_SITE_<language>.HTML` (note the leading underscore, which means it will not be overwritten by a soyMAIL update, and the language component). If this file is found during help composition it is appended to the primary help page (that obtained in help by using the [help] button) and is intended for providing site information and/or links of local relevance.

5.3 User

The options button in the main menu provides access to user options settings.

6 soyMAIL User Files

soyMAIL creates and maintains several files in the same directory as the users MAIL.MAI file. The names are all prefixed with `SOYMAIL_` (an underscore). Needless-to-say these files should not be deleted or edited without good reason and care. Doing so potentially disrupts the users soyMAIL environment. Some, all, multiple instances, or none of these may be present at any given time.

File Name	Purpose
SOYMAIL_XXXXXXXXXXXXX.ATT	Binary content attachment file either saved from a message or uploaded via the composition page.
SOYMAIL_CONTACTS.LDIF	LDAP Data Interchange Format (LDIF) file containing users contact and address information.
SOYMAIL_DRAFT.TXT	Not used with v1.3 or later.
SOYMAIL_OPTIONS.TXT	Plain text configuration file storing the users option settings.
SOYMAIL_SIGNATURE.TXT	Plain text file containing the soyMAIL-specific signature.
SOYMAIL_YOUGOTMAIL.WAV	Binary audio file containing users optional "you've got mail" audible alert.

Extracted Files

The soyMAIL message read attachment (message part) extract button (where not VMS-occluded) allows message components to be written to the VMS user account home directory. All files created have the name prefixed with `SOYMAIL-` (a hyphen) and the rest generated from the attachment/part name. If that part name contains a file type (e.g. .GIF, .PDF, .TXT) then that is used in the name. If it does not contain such a type then soyMAIL attempts to generate one from the MIME content-type.

MIME content-type	VMS Record Format	Generated Extension
text/plain	stream-LF	.TXT
text/html	stream-LF	.HTML
text/..	stream-LF	<i>none</i>
message/rfc822	stream-LF	.LIS
message/..	stream-LF	<i>none</i>
<i>anything else</i>	fixed 512 bytes	<i>none</i>

7 Miscellaneous Topics

Miscellaneous soyMAIL features and other topics.

7.1 *Run-time Problem Solving*

When an error is reported, either fatal or in the status panel, the source code module name and line number of the reporting point is included as an HTML comment (the page source needs to be opened and searched) to assist in locating and rectifying issues.

```
<!-- ***** REPORTING MODULE:LINE IS MESSAGE:2822 ***** -->
```

Please include this information when reporting problems.

Defining the system-level logical name SOYMAIL\$WATCH to either TRUE or the IP address of the client to be observed provides a plain-text report designed to assist in solving configuration or software issues with soyMAIL.

7.2 *Inconsistent State Data*

To prevent data corruption and inconsistent behaviours soyMAIL performs integrity checks on the state data it propagates from request to request. It is possible for a user session spanning a soyMAIL update (version release) to see the following error status message.

```
Inconsistent state data (version). Restarting session.
```

This is of no concern. The change in version has been noted by the software and to prevent any potential inconsistencies in data structures causing subtle or gross problems it has reinitialized the state data resulting in an effectively empty session. The user should just reopen (not refresh/reload) the particular page.

The second variation of this message is a little more concerning.

```
Inconsistent state data (corruption). Restarting session.
```

soyMAIL maintains a hash of the state data which is propagated with it. The hash is recalculated and compared at the next request. This error reports the comparison failed and indicates data corruption in the request state. The session is effectively emptied as a precautionary measure. Instances of this message should be very rare and if persistent carefully investigated.

7.3 *Site Contact / Mailing Lists*

The logical name SOYMAIL_CONTACT_LIST can be used to specify a logical list of contact lists (in addition to any personal contacts). This functions as a multi-value logical name with each value being the logical name for, or actual file specification of, a soyMAIL format contacts list or a VMS-style (each line in the file contains a single address) mailing list.

soyMAIL

For example:

```
$ DEFINE /SYSTEM SOYMAIL_CONTACT_LIST -  
  ALL_USERS_LIST, GROUP1_USERS_LIST, GROUP2_USERS_LIST, GROUP3_USERS_LIST, -  
  EXTERNAL_LIST, MAILING_LIST_LIST
```

Where ALL_USERS_LIST is a VMS-style, so are GROUP1_USERS_LIST and GROUP2_USERS_LIST. MAILING_LIST_LIST is a file containing a VMS-style mailing list of the mailing lists supported by the system. For the above it might contain five lines with:

```
ALL_USERS_LIST  
GROUP1_USERS_LIST  
GROUP2_USERS_LIST  
GROUP3_USERS_LIST  
EXTERNAL_LIST
```

These can then be selected to mail to the entire list.

7.4 General Access To Help

Context-sensitive help is available to authenticated users accessing private mail. A page built from the consolidated help information (and distinctly resembling the *print* page for context-sensitive help) is available for unauthenticated (general) access. Use the following URL to access this page.

```
http://the.host.name/cgi-bin/soymail?help
```