VISUALMAIL 3.0

System Administrator Guide

CHAPTER 1:

Introduction to VisualMail

About VisualMail

The inability to access your e-mail while you are away, was until now, one of the major problems. Configuring any conventional mail package to access your mailbox is annoying and sometimes impossible if your server is protected by firewalls, and why make your network unsecure disabling such protection in order to access your mail remotely?. Free Mail services such as Hotmail, force you to have an additional e-mail address and save your confidential mail on an untrusted server, something not very useful for most people.
That is why VisualMail was created, a fully featured web based mail software designed to enable any user to access his/her POP3/IMAP mailbox using a standard browser or web terminal.
With VisualMail there is no need to worry about configurations, firewalls, privacy or where you are, because it was designed with the security in mind, it runs on your server, you access it via web and you can customize it to fit your needs.
VisualMail supports unlimited users and works with UNIX mail servers as well as Microsoft Exchange and Lotus Notes Domino mail servers.

Choosing the right VisualMail version

VisualMail is available in two versions, "Standard version" and "Enterprise version". It's important that you understand the functional differences between the two versions in order to choose the more suitable one for your needs.

The standard version is a gateway to any POP3/IMAP4 mailbox, that fetches messages from a mail server and displays them in HTML format.
This version is more suitable for ISPs or companies where the users want to use VisualMail as as complement to their conventional software, for example, an user that is travelling and wants to check his email via web and then, when he is back home, downloads his mails using his conventional e-mail package.

The enterprise version is a fully featured mail program, that fetches messages from a mail server and stores them into an internal database, just like as Hotmail does. This version is more suitable for free mail providers (like Hotmail) or companies where the users want to use VisualMail as their conventional e-mail software, for example, an user that stores his mails on the VisualMail server and has access to them using a browser.

Features

Standard version features:

- POP3 / IMAP4 server support.
- SMTP server support.
- Multi-Language support.
- HTX files: customizable HTML files with HTX language.
- Built-in security: object access validation.
- MIME/UUENCODE support.
- ISO-8859-1 Quoted printable and Base64 embedded in headers support.
- Attachment upload support: multiple attachment sending support.
- Cross Server or standard passwd / shadow validation.
- Syslog logging: configurable logging levels.
- User access locking: banned users list.
- Automatic disconnection.
- Frames and Non-Frames support.
- On-Line help: on-line help available on all screens.
- Address Book.
- 5 additional mail servers: user configurable additional mail servers.
- Configurable mails per page listings.
- URL Hot-Linking.

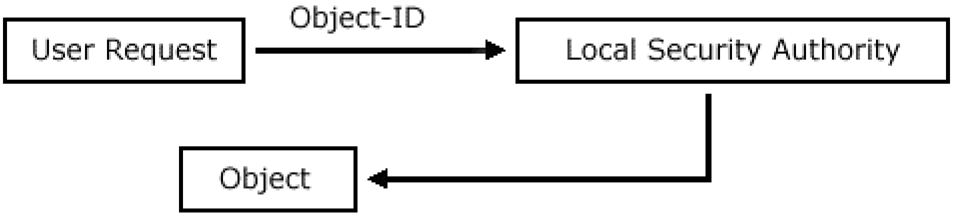
- Personal Configuration: Signature, Full Name, Organization, etc.
- Message sorting: by name, subject, date and size.
- Configurable Main Mail Server.

Enterprise version features:

- POP3 / IMAP4 server support.
- SMTP server support.
- Multi-Language support.
- HTX files: customizable HTML files with HTX language.
- Built-in security: object access validation.
- MIME/UUENCODE support.
- ISO-8859-1 Quoted printable and Base64 embedded in headers support.
- Attachment upload support: multiple attachment sending support.
- Cross Server or standard passwd / shadow validation.
- Syslog logging: configurable logging levels.
- User access locking: banned users list.
- Automatic disconnection.
- Frames and Non-Frames support.
- On-Line help: on-line help available on all screens.
- Address Book.
- 5 additional mail servers: user configurable additional mail servers.
- Configurable mails per page listings.
- URL Hot-Linking.
- Personal Configuration: Signature, Full Name, Organization, etc.
- Message sorting: by name, subject, date and size.
- Folders support.
- Filters support: users can configure filters to deliver messages to particular folders when a string is matched.
- Disk Quota: configurable disk space management.
- Enhanced databases: Fast internal database, dynamic space allocation and database locking.
- Database cleaning utility: Deleted messages cleaning and database fixing utility.

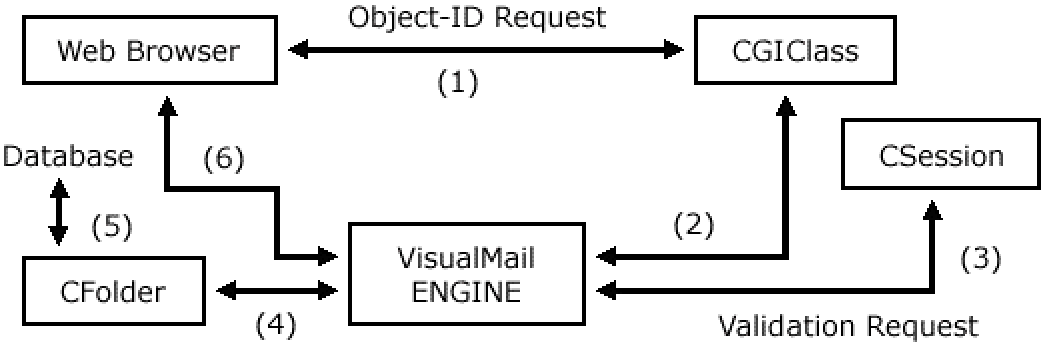
Architecture Overview

VisualMail is written in C++ and its modular architecture makes it a scalable and multiplatform solution easily adaptable to new technologies.
All the data is handled by VisualMail as an object. All the objects are referenced with an Object identifier (also called Session identifier). Before processing an object, VisualMail checks the Object-ID against the Local Security Authority (called CSession) that validates if the user is logged in, if the session has not expired and if the user is allowed to gain access to the object.



The object accessed could be an attachment, an email address, a message or a particular action to perform.

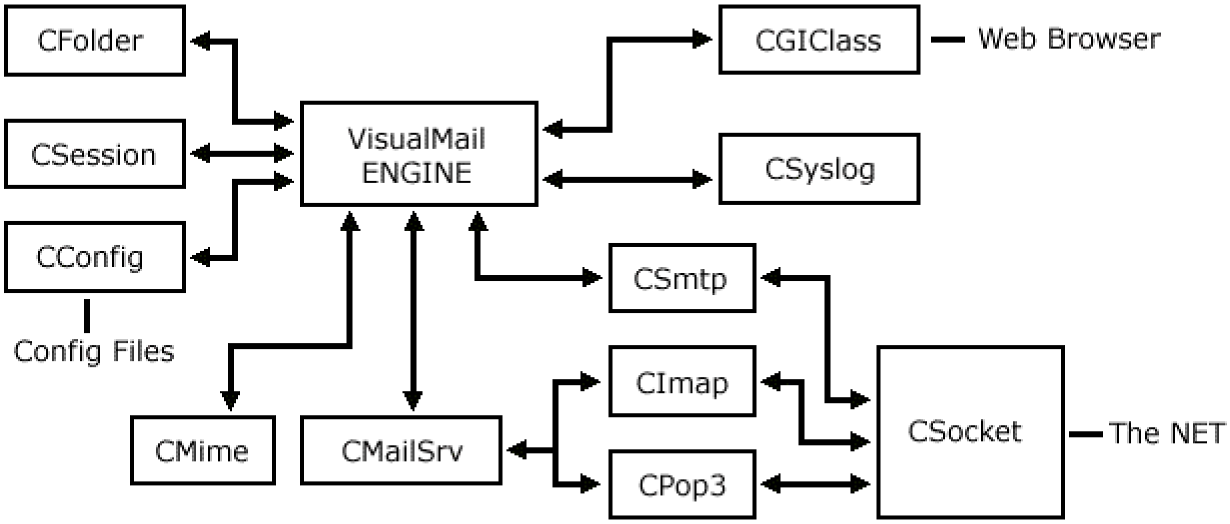
All the internal data flow between the different modules is managed by the VisualMail Engine, these modules handle specific functions like interfacing with users or servers, and performing particular actions. Let's take the example of an user downloading an attachment:



1. The user requests the attachment by sending its corresponding Object-ID.
2. The **CGIClass** receives the request from the browser and converts it to a format understandable by the Engine.
3. Once the object request is received by the Engine, it is validated against the Local Security Authority (**CSession**).
4. If the object is successfully validated, the request is sent to the Folder Manager (**CFolder**).
5. The Folder Manager accesses the requested file and sends it back to the Engine for further processing.
6. The Engine receives the file and sends it to the requesting user.

All the objects are requested and validated using the same method, but they are handled using a different handler (**CFolder** in this example).

The most important objects are:



CGIClass: Handles the interaction with the user, it receives the data posted by the browser and processes it.

CSmtp: Handles the communication with the SMTP servers.

CPop3: Handles the communication with the POP3 servers.

CImap: Handles the communication with the IMAP servers.

CSocket: Handles the TCP connections used by the **CSmtp**, **CPop3** and **CImap** modules.

CMailSrv: Interfaces the **CPop3** and **CImap** modules and processes RFC-822 headers.

CMime: Handles MIME, UUENCODE and ISO-8859 encoding and decoding.

CConfig: Handles the system and user configuration.

CSession: Handles Object-ID requests and validation.

CSyslog: Handles logging.

CFolder: Handles database object access in the enterprise version.

CVFolder: Handles database object access in the standard version.

CHAPTER 2:

Installing VisualMail

Installing VisualMail Enterprise

1. Unpack the VisualMail distribution file.

   ```
   tar zxvf visualmail-3.0.tar.gz
   ```

2. cd to the distribution directory.

   ```
   cd visualmail-3.0/
   ```

3. Become root.

   ```
   su root
   ```

4. Run the setup program.

   ```
   ./setup
   ```

   Then you will have to enter some directories information:

   ```
   Installation directory [/usr/local/visualmail]:
   ```

   The Installation directory is where all the VisualMail file will be installed.

   ```
   Web server cgi-bin directory [/var/lib/httpd/cgi-bin]:
   ```

   The cgi-bin directory is the cgi executable directory of your web server software where the VisualMail executable file will be stored.

   ```
   Web server document root directory [/var/lib/httpd/htdocs]:
   ```

   The document root directory is the html root directory of your web server software where the VisualMail gif files will be stored.

5. Edit the **/etc/visualmail.conf** file in order to match your system configuration. For more information about the visualmail.conf file parameters, read the visualmail.conf section in this guide.

6. Add an entry to your crontab file to run the 'dbutil' program every night. The 'dbutil' handles all the database maintenance functions.

   ```
   crontab -e
   ```

   and add the following line (example):

   ```
   00 05 * * *   <your_install_directory>/bin/dbutil >/dev/null
   ```

   The 'dbutil' program must be executed as root in order to open all the databases.

7. Open your browser and try VisualMail.

Installing VisualMail Standard

1. Unpack the VisualMail distribution file.

   ```
   tar zxvf visualmail-3.0.tar.gz
   ```

2. cd to the distribution directory.

   ```
   cd visualmail-3.0/
   ```

3. Become root.

   ```
   su root
   ```

4. Run the setup program.

   ```
   ./setup
   ```

   Then you will have to enter some directories information:

   ```
   Installation directory [/usr/local/visualmail]:
   ```

   The Installation directory is where all the VisualMail file will be installed.

   ```
   Web server cgi-bin directory [/var/lib/httpd/cgi-bin]:
   ```

   The cgi-bin directory is the cgi executable directory of your web server software where the VisualMail executable file will be stored.

   ```
   Web server document root directory [/var/lib/httpd/htdocs]:
   ```

   The document root directory is the html root directory of your web server software where the VisualMail gif files will be stored.

5. Edit the **/etc/visualmail.conf** file in order to match your system configuration. For more information about the visualmail.conf file parameters, read the visualmail.conf section in this guide.

6. Open your browser and try VisualMail.

CHAPTER 3:

Configuring VisualMail

The VisualMail configuration file visualmail.conf

The visualmail.conf file contains all the configuration parameters of VisualMail and is located in the /etc directory.

- **TemplateDir = <path>**

  Is the directory where the HTX files are placed. Read the HTX section for more information about the HTX files.
  This parameter applies to both versions.

- **HomeDir = <path>**

  Is the directory where VisualMail saves the configuration, expiration data, and folders of each user.
  This parameter applies to both versions.

- **LangFile = <path_to_file>**

  Is the file that contains all the VisualMail messages listed one per line.
  This parameter applies to both versions.

- **TmpDir = <path>**

  Is the temporary directory where VisualMail saves temporary data. (eg. /tmp)
  This parameter applies to both versions.

- **URLVisualMail = <string>**

  Is the URL of the VisualMail executable file. This parameter is used to compose HTML files.
  (eg. /cgi-bin/visualmail)
  This parameter applies to both versions.

- **BannedUsers = <path_to_file>**

  Is the file that contains all the users that can't access VisualMail listed one per line.
  This parameter applies to both versions.

- **Validation = <string>**

  Is the validation method used by VisualMail.

  VisualMail supports two validation methods:

  1. **Passwd/Shadow validation:** Usernames and passwords are validated against the system's passwd or shadow file.
  2. **Cross Server validation:** Usernames and passwords are validated against the Main Mail Server.

  You can specify here:

  - **Cross Server:** use the Main Mail server to validate user/passwords.
  - **<path_to_file>:** use <file> to validate username and passwords, where <file> is a passwd compatible format file. (eg. passwd or shadow)

  This parameter applies to both versions.

- **SMTPAddress = <string>**

  Is the domain name or IP address of the SMTP server used to send mails.
  This parameter applies to both versions.

- **SMTPPort = <number>**

  Is the port number of the SMTP server used to send mails.
  This parameter applies to both versions.

- **MMSAddreess = <string>**

  Is the domain name or IP address of the Main Mail Server. This is the server accessed by default.
  This parameter applies to both versions.

- **MMSPort = <number>**

  Is the port number of the Main Mail Server, usually 110 for POP3 and 143 for IMAP4 servers.
  This parameter applies to both versions.

- **MMSType = <string>**

  Is the Main Mail Server type.

  You can specify here:

  POP3: for POP3 server.
  IMAP: for IMAP2 or IMAP4 servers.

  This parameter applies to both versions.

- **MMSLeaveOnSrv = <boolean>**

  If VisualMail should leave the mails on the Main Mail Server once downloaded.
  You can specify here either yes or no.
  This parameter applies to the enterprise version.

- **DefDomain = <string>**

  Is the default domain name used to compose user's e-mail addresses.
  This parameter applies to both versions.

- **AutoLogout = <number>**

  Is the amount of time that an user can be connected without activity or 0 to disable AutoLogout.
  This parameter applies to both versions.

- **AttachLimit = <number>**

  Is the maximum attachment file size allowed to upload, expressed in kilobytes or 0 to disable the attach limit.
  This parameter applies to the enterprise version.

- **QuotaDisk = <number>**

Is the maximum disk space that an user can use for storing his/her mails, expressed in megabytes or 0 to disable disk quota.
This parameter applies to both versions.

- **LogLevel = <number>**

  This parameter specifies the logging level used by VisualMail, the entries are recorded using the syslog facility.

  Levels are:

  0: Disable logging.
  1: Logs logins and logouts.
  2: Logs logins, logouts and several errors.
  3: Logs logins, logouts and all the errors.
  4: Logs everything.

The Banned Users file

The banned.cfg file contains a list of users that can't have access to VisualMail, one per line. Is suggested that you add any root users of your system to this list in order to avoid force brute password cracking.

The Language file

The language.cfg file contains all the messages displayed by VisualMail listed one per line. Modifying this file, you can customize date formatting options, error messages and some default texts.
The language file is common to both versions of VisualMail. Don't delete or add any line(s) of the language file, or you will receive error messages.

The date formatting macros are the same used by strftime():

%a The abbreviated weekday name according to the current locale.

%A The full weekday name according to the current locale.

%b The abbreviated month name according to the current locale.

%B The full month name according to the current locale.

%c The preferred date and time representation for the current locale.

%d The day of the month as a decimal number (range 01 to 31).

%H The hour as a decimal number using a 24-hour clock (range 00 to 23).

%I The hour as a decimal number using a 12-hour clock (range 01 to 12).

%j The day of the year as a decimal number (range 001 to 366).

%m The month as a decimal number (range 01 to 12).

%M The minute as a decimal number.

%p Either Am' or Pm' according to the given time value, or the corresponding strings for the current locale.

%S The second as a decimal number.

%U The week number of the current year as a decimal number, starting with the first Sunday as the first day of the first week.

%W The week number of the current year as a decimal number, starting with the first Monday as the first day of the first week.

%w The day of the week as a decimal, Sunday being 0.

%x The preferred date representation for the current locale without the time.

%X The preferred time representation for the current locale without the date.

%y The year as a decimal number without a century (range 00 to 99).

%Y The year as a decimal number including the century.

%Z The time zone or name or abbreviation.

The HTX Language

VisualMail creates all the HTML files using a languange called HTX.
The HTXs files contain a mini language used to place dynamic text and objects in the HTML that is displayed to the end-user.
The structure of the VisualMail HTXs is similar to the HTXs used by Microsoft Internet Information Server.
This language is very simple, it has only 'if' and 'while' control commands and tags used to print dynamic information and HTML objects.

Let's view an example of the HTX language:

```
1:<H1>Welcome <% UserFullName %> to VisualMail!</H1><BR>
2:<% if HaveMail %>
3:<B><U>You have mail!</U></B><BR>
4:<% while GetMail %>
5:<HR>
6:  Message from: <% MsgFrom %><BR>
7:  Subject: <% MsgSubject %><BR>
8:  Size: <% MsgSize %><BR>
9:  Date: <% MsgSize %><BR>
10: <% while end %>
11: <% else %>
12: <B><U>You don't have mail.</U></B><BR>
13: <% end if %>
```

Line 1 prints the Full Name of the current logged in user.
Line 2 checks if the user has mail on his mailbox, and if this condition is true the HTX is parsed until the **<% else %>** directive is found, otherwise the HTX is parsed from the **<% else %>** to the **<% end if %>** directive.
Line 4 is parsed from the **<% while %>** to the **<% end while %>** as many times as messages in the mailbox.
Lines 6 to 9 print information about each message.

Each HTX page has different commands, but there are some global commands that can be referenced from any page.

The following table describes the function of each HTX file:

File	Name	Description
login.htx	Login screen	Where the login is performed.
error.htx	Error screen	Displayed when an error occurs.
index.htx	Index Screen	Displayed after the user logs in, usually this screen is used to specify frames information.
optbar.htx	Option Bar Screen	Displayed as a frame, contains a toolbar. This screen doesn’t contain additional HTX commands.
inbox.htx	InBox Screen	Where the messages are listed.
viewmsg.htx	Message View Screen	Where a message is viewed.
options.htx	Options Screen	Where the user selects which section to configure.
filterconf.htx	Filters Configuration Screen	Where the filters are configured. This screen is only available in the enterprise version.
persconf.htx	Personal Configuration Screen	Where the personal options are configured.
servconf.htx	Server Configuration Screen	Where the additional mail servers are configured.
address.htx	Address Book Screen	Where the e-mail addresses and mailing lists are maintained.

folder.htx	Folder Screen	Where the folders are maintained. This screen is only available in the enterprise version.
signconf.htx	Signature Configuration Screen	Where the signature is configured.
compose.htx	Message Compose Screen	Where the messages are composed.
logout.htx	Logout Screen	Displayed when the user logout.

As you can see this is a very simple language but very helpful to customize VisualMail and change default behaviors.
If you want to write your own HTX files, use as a guide the examples that come with VisualMail and read the Appendix A (HTX command reference).

CHAPTER 4:

Troubleshooting VisualMail

Troubleshooting guide

1. Check if visualmail has SUID/SGID permissions:

   ```
   # ls -la

   -rwsr-sr-x   1 root     root       569012 Jan 15  1997 visualmail
   ```

   If visualmail is not SUID/SGID, run the commands:

   ```
   chown root visualmail
   chgrp root visualmail
   chmod 6755 visualmail
   ```

2. Check all the paths to files and directories specified in the */etc/visualmail.conf*.

3. Check if there is enough disk space on the temporary directory and on the visualmail home directory.

4. Check if the language file is complete.

5. Check if all the HTX files are present in the HTX directory.

6. Check if the URL of the visualmail executable file is correctly specified in the */etc/visualmail.conf*.

7. Check if the /images directory is installed in your html root directory and can be accessed from the web browser.

8. Check the SMTP server address and port number.

9. Check the Main Mail server address, port number and type.

Common Problems and its solutions

Problem: Performing actions that take too much time (like downloading a very big mailbox), the user always receives after a constant amout of time a "the server returned an invalid or unrecognized" error message.

Cause: Web server configuration issue.

Solution: Check the ScriptTimeout parameter in your web server's config file, by default most web server software kills the cgi application if no reponse is received for more than x time.

Problem: VisualMail returns a "Incomplete config file" error message.

Cause: Missing parameter in *visualmail.conf*

Solution: Check if the VisualMail configuration file *visualmail.conf* is located in the /etc directory and contains all the required parameters.

Problem: VisualMail returns a "Error reading language file" message.

Cause: VisualMail can’t access the language file.

Solution: Check if the language file specified in the *visualmail.conf* file is present and can be readed from VisualMail.

Problem: VisualMail returns a "Error processing <file>.htx file" message.

Cause: <file>.htx can’t be readed by VisualMail.

Solution: Check if the HTX directory specified in the *visualmail.conf* file is present and can be readed from VisualMail. Also check is the <file>.htx file is present and can be readed from VisualMail.

Problem: VisualMail returns a "Error parsing HTX file" message.

Cause: The HTX file selected can’t be accessed from VisualMail or contains incorrect HTX commands.

Soultion: Check if the HTX path specified in the *visualmail.conf* file is present and can be accessed from VisualMail. Also check if the HTX file accessed contains any incorrect HTX command.

Problem: The images appear as broken links.

Cause: The /images directory can’t be accessed from the web browser.

Solution: Check if the /images directory is copied correctly in your web server html root directory.

Problem: The messages deleted are not cleaned at nights (enterprise version)

Cause: The dbutil program is not installed correctly in the cron file.

Solution: Read the installation procedure described in chapter 2.

APPENDIX A:

HTX Command Reference

HTX Commands reference

Global Commands

<% UserName %>

Prints the username of the current logged in user.

<% UserFullName %>

Prints the Full Name of the current logged in user.

<% UserCompany %>

Prints the Company name of the current logged in user.

<% UserEMail %>

Prints the e-mail address of the current logged in user.

<% DefaultDomain %>

Default domain name specified in *visualmail.conf*

<% LicenseName %>

Prints the licensed to name specified in *visualmail.conf*

<% AutoLogout %>

Automatic Logout time specified in *visualmail.conf*

<% StartForm %>

Starts a HTML form.

<% EndForm %>

Ends a HTML form.

<% HiddenVar %>

Prints all the hidden vars needed by VisualMail.

<% Goto Compose %>

Prints the Object-ID of the compose screen.

<% Goto Inbox %>

Prints the Object-ID of the Inbox screen.

<% Goto Address %>

Prints the Object-ID of the Address Book screen.

<% Goto Config %>

Prints the Object-ID of the configuration screen.

<% Goto Folder %>

Prints the Object-ID of the Folder screen.

<% Goto Logout %>

Prints the Object-ID of the Logout screen.

<% Goto Accept %>

Prints the Object-ID of the Accept command.

<% If HaveNoFrames %>

If the user is accessing VisualMail without frames, the HTX is parsed until an *<% else %>* or an *<% end if %>* is found.

error.htx commands

<% ErrorMsg %>

Prints the error message.

login.htx commands

<% LoginBox = x, y %>

Prints the Login text box, with x chars of length and y characters of max length.

<% PassBox = x, y %>

Prints the Password text box, with x chars of length and y characters of max length.

<% FramesCheck %>

Prints the select frames option check.

<% NoFramesCheck %>

Prints the select no frames option check.

index.htx commands

<% Goto OptionBar %>

Prints the Object-ID of the Option Bar screen.

<% Goto InboxNoFrm %>

Prints the Object-ID of the Inbox screen without frames support.

inbox.htx commands

<% If HaveMail %>

If the user has mail, the HTX is parsed until an *<% else %>* or an *<% endif %>* is found.

<% If HaveAttach %>

If a message has an attach, the HTX is parsed until an *<% else %>* or an *<% endif %>* is found.

<% If Urgent %>

If a message is urgent, the HTX is parsed until an *<% else %>* or an *<% endif %>* is found.

<% While GetMail %>

The HTX is parsed until the *<% while end %>* command is found while the user has mails.

<% Goto SortByName %>

Prints the Object-ID of the Sort by Name command.

<% Goto SortBySubj %>

Prints the Object-ID of the Sort by Subject command.

<% Goto SortBySize %>

Prints the Object-ID of the Sort by Size command.

<% Goto SortByDate %>

Prints the Object-ID of the Sort by Date command.

<% Goto MsgCheck %>

Prints the CheckBox used to select a message.

<% Goto CheckMail %>

Prints the Object-ID of the Check Mail command.
This command only works in the enterprise version.

<% MsgFrom %>

Prints the sender name of the message.

<% MsgSubject %>

Prints the Subject of the message.

<% MsgSize %>

Prints the message’s Size.

<% MsgDate %>

Prints the message’s sent date.

<% DFolderList = x %>

Prints the destination folders list with x lines of size.
This command only works in the enterprise version.

<% CurrentPage %>

Prints the current page number.

<% MaxPage %>

Prints the max page number.

<% FolderList = x %>

Prints the folders list, with x lines of size.
This command only works in the enterprise version.

<% ServerList = x %>

Prints the server list, with x lines of size.
This command only works in the standard version.

<% CurrentFolder %>

Prints the current folder name.
This command only works in the enterprise version.

<% CurrentServer %>

Prints the current server name.
This command only works in the standard version.

viewmsg.htx commands

<% Goto Reply %>

Prints the Object-ID of the Message Reply Command.

<% Goto ReplyAll %>

Prints the Object-ID of the Message Reply All Command.

<% Goto Forward %>

Prints the Object-ID of the Message Forward Command.

<% Goto Delete %>

Prints the Object-ID of the Message Delete Command.

<% Goto Address %>

Prints the Object-ID of the Add to Address Book command.

<% MsgFrom %>

Prints the sender of the message.

<% MsgTo %>

Prints the recipients of the message.

<% MsgCc %>

Prints the carbon copy recipients of the message.

<% MsgSubject %>

Prints the subject of the message.

<% MsgDate %>

Prints the sent date of the message.

<% MsgBody %>

Prints the body of the message.

<% If HaveAttach %>

If a message has attachments, the HTX is parsed until an *<% else %>* or an *<% endif %>* is found.

<% While GetAttach %>

The HTX is parsed until the *<% while end %>* command is found while the message has attachments.

<% MsgAttach %>

Prints the Object-ID of the attachment.

options.htx commands

<% Goto FilterConf %>

Prints the Object-ID of the Filters Configuration screen.
This command only works in the enterprise version.

<% Goto SignConf %>

Prints the Object-ID of the Signature Configuration screen.

<% Goto PersonalConf %>

Prints the Object-ID of the Personal Configuration screen.

<% Goto AdvancedConf %>

Prints the Object-ID of the Advanced Configuration screen.

filterconf.htx commands

<% ItemNumber %>

Prints the filter number.

<% StatusCheck %>

Prints the filter status check box.

<% WhereList = x %>

Prints the list of "where to find the string" with x lines of size.

<% ConditionList = x %>

Prints the list of conditions to match with x lines of size.

<% MatchString = x, y %>

Prints the match string text box with x chars of size and a max length of y chars.

<% FolderList = x %>

Prints the list of folders with x lines of size.

<% While GetFilter %>

The HTX is parsed until the *<% while end %>* command is found while the message has filters to show.

persconf.htx commands

<% FullNameBox = x, y %>

Prints the Full Name text box with x chars of length and y chars of max length.

<% EMailBox = x, y %>

Prints the E-Mail text box with x chars of length and y chars of max length.

<% CompanyBox = x, y %>

Prints the Company Name text box with x chars of length and y chars of max length.

<% PageSizeBox = x, y %>

Prints the Page Size text box with x chars of length and y chars of max length.

signconf.htx commands

<% SignBox = x, y %>

Prints a signature text area with x rows and y columns.

servconf.htx commands

<% StatusCheck %>

Prints the server status check box.

<% ItemNumber %>

Prints the server number.

<% LoginBox = x, y %>

Prints the login name text box with x chars of length and y chars of maxlength.

<% PassBox = x, y %>

Prints the login pass text box with x chars of length and y chars of maxlength.

<% ServerBox = x, y %>

Prints the server text box with x chars of length and y chars of maxlength.

<% PortBox = x, y %>

Prints the port text box with x chars of length and y chars of maxlength.

<% Pop3Check %>

Prints the POP3 server option box.

<% ImapCheck %>

Prints the IMAP server option box.

<% LeaveMailCheck %>

Prints the Leave Mail on Server check box.
This command only works in the enterprise version.

<% While GetServer %>

The HTX is parsed until the *<% while end %>* command is found while the message has servers to show.

address.htx commands

<% Goto Cancel %>

Prints the Object-ID of the Cancel command.

<% AddressList = x %>

Prints the list of addresses with x lines of size.

<% FullNameBox = x, y %>

Prints the full name text box with x chars of length and y chars of maxlength.

<% EMailBox = x, y %>

Prints the E-Mail box with x chars of length and y chars of length.

<% If Modify %>

If the user is in modify mode, the HTX is parsed until an *<% else %>* or an *<% endif %>* is found.

folder.htx commands

<% Goto Cancel %>

Prints the Object-ID of the Cancel command.

<% FolderList = x %>

Prints the list of folders with x lines of size.

<% FolderBox = x, y %>

Prints the folder text box with x chars of length and y chars of maxlength.

<% If Modify %>

If the user is in modify mode, the HTX is parsed until an *<% else %>* or an *<% endif %>* is found.

compose.htx commands

<% MsgTo = x, y %>

Prints the Message To text box with x chars of length and y chars of maxlength.

<% MsgCc = x, y %>

Prints the Message Cc text box with x chars of length and y chars of maxlength.

<% MsgBcc = x, y %>

Prints the Message Bcc text box with x chars of length and y chars of maxlength.

<% MsgSubject = x, y %>

Prints the Message Subject text box with x chars of length and y chars of maxlength.

<% MsgFiles = x, y %>

Prints the Attachments text box with x chars of length and y chars of maxlength.

<% PriorityLevel = x %>

Prints the priority levels list with x lines of size.

<% AddressList = x %>

Prints the Address List with x lines of size.

<% MsgBody = x, y %>

Prints the Message Body text area with x rows and y columns.

<% AttachBox = x %>

Prints the Attach text box with x chars of length.

<% SignCheck %>

Prints the Add Signature check box.

<% AddSentCheck %>

Prints the Add to sent items check box.
This command only works in the enterprise version.