Contents

Virus Protection

The Mailing List Plug-in

Shared Address Book

IMAP

WebMail

WorkgroupMail Plug-ins

Developer Information

WorkgroupMail API

Overview

WorkgroupMail is a fully featured mail server. It is designed to handle the e-mail needs of any sized organization. Primarily, it operates as a standalone mail server for organizations that host their own e-mail. Furthermore, it is capable of sending and receiving mail from a variety of ISPs (Internet Service Providers) for organizations that do not host their own e-mail and rely on an ISP for delivery of their e-mail.

Centralization of e-mail provides greater control and security over the communications flowing in and out of your organization. WorkgroupMail provides the ability to perform server based antivirus checking and content filtering on all inbound and outbound messages.

Different organizations send and receive mail in different ways. For example, some employ a variety of separate POP3 accounts, hosted by one or more ISPs. Some use a single multiple drop POP3 account. Smaller companies sometimes have only a single address POP3 account. Larger organizations, which host their own e-mail, send and receive their e-mail directly, independent of an ISP, using SMTP. Some organizations choose to centralise the storage of e-mail, requiring staff to collect their e-mail using IMAP. WorkgroupMail provides a solution for all these scenarios.

WorkgroupMail can run on Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000 and Windows XP. It has no special hardware requirements over and above the mimimum requirements for the particular operating system.

This user guide describes in detail the procedure for installing and configuring the software, specifically for your own environment.

Installing WorkgroupMail

The Welcome Page

WorkgroupMail is packaged as both a ZIP file and self extracting EXE file. If you download the ZIP file you will need to extract the files to a temporary directory using WinZip (a popular file compression program downloadable from http://www.winzip.com) and run **Setup.exe**. If you download the EXE file then you can simply run this file to begin the Setup program.

The Setup program displays a wizard which will guide you through the installation of WorkgroupMail. The first page of the wizard is the Welcome page. If you want step by step instructions on how to configure WorkgroupMail and how to configure your mail clients, then click on the "Click here for a step by step guide" link in this page. Press **Next** to continue.

The next page requires you to read the license agreement and agree to the terms. Press the **Accept** button to agree to the terms and to continue.

If you are running Windows NT, Windows 2000 or Windows XP, the next page provides you with the choice of running WorkgroupMail as an executable program or as a service. Select the appropriate option and press **Next**.

The Select Folder Page

The next page lets you choose the installation folder and the data folder. By default, WorkgroupMail will be installed to **c:\program files\workgroupmail** and the data files will be located in **c:\program files\workgroupmail\data**. If you wish to install WorkgroupMail to a different folder or disk, then use the **Browse…** button to locate the appropriate folder.

Note: If you have previously installed WorkgroupMail to the folder specified in this page or you have chosen to upgrade an older version of WorkgroupMail then you will skip to the Summary page. If you wish to re-install completely to the same folder then you must remove the file **Program Files/WorkgroupMail/data/wmdata.dat** and re-run Setup.

The Configuration Page

The Configuration page lets you choose how you wish to use WorkgroupMail.

WorkgroupMail may be configured for use with an ISP, for use in a self-hosted mail domain or for internal mail only. Choose the option that is appropriate to your organization and press **Next**.

The Details Page

The Details page requires you to enter your organization name, your name and your e-mail address.

Ensure that you enter your e-mail address accurately, since Setup will use this to configure certain settings in WorkgroupMail.

ISP Page (ISP Configuration)

If you selected **Standard** ISP **Based Mail Server** in the configuration page, the ISP page is shown next. This page requires you to enter the name of your ISP and the address of the ISP's Incoming and Outgoing mail servers.

Enter the appropriate information and press **Next**.

Account Information Page (ISP Configuration)

The next page requires you to enter the account name and password that you use to gain access to your ISP's POP3 server for receiving mail.

You should specify whether this account collects mail for everyone or just for you by selecting the appropriate radio button. Press **Next** to continue.

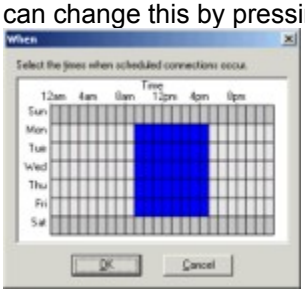
Internet Connection Page (ISP Configuration)

This page lets you specify whether you connect to the Internet directly or by dialing up to connect.

Select the appropriate radio button. If you select the **Dial-up to connect** radio button, you should specify a dial-up service to use. Press **Next** to continue.

The Connection Page (ISP Configuration)

The next page lets you specify when to connect to the ISP in order to send and receive mail. If you wish to connect when the WorkgroupMail program is started, tick the **Connect on startup** tick box. If you wish to connect when there are more than *n* messages waiting to be sent, regardless of any other connection options, then tick the **Connect when *n* messages to send** and enter the appropriate number of messages. If you wish to connect on a regular basis, then tick the **Connect every *n* minutes** and enter the appropriate interval in minutes. If you tick this tick box then, by default, WorkgroupMail will connect every *n* minutes, only between the hours of 9pm and 5pm, Monday to Friday and not at weekends. You can change this by pressing the **When…** button.



The When dialog box shows a grid with weekdays along one axis and time along the other. The blue areas of the grid represent the times when WorkgroupMail will connect to the ISP every *n* minutes in order to send and receive mail. You can change the color of the grid by clicking the left mouse button and dragging the mouse over the appropriate areas. Press **OK** to save your changes. Press **Next** to continue. The next page is the Summary page.

The Details Page (Enterprise Configuration)

If you selected **Enterprise level mail server** in the configuration page, the next page shown after the Details page is the DNS Servers page . This page requires you to specify a primary (and optionally secondary) DNS server. This should be a DNS server on the Internet.

Pressing **Next** will show the Summary page.

The Summary Page

The last page is the Summary page which summarizes the choices you have made in the wizard. Press **Finish** to begin the installation. When the installation is complete, the WorkgroupMail program folder is shown.

The WorkgroupMail Administrator

The WorkgroupMail Administrator lets you further configure WorkgroupMail after installation and also manage other administrative tasks. Using the WorkgroupMail administrator, you can add additional local users, virtual mailboxes, domains and ISPs. You can configure the server-based antivirus checking. You can define the content filtering rules and administer the shared folders and address book. You can also monitor the flow of messages in and out of the organization.

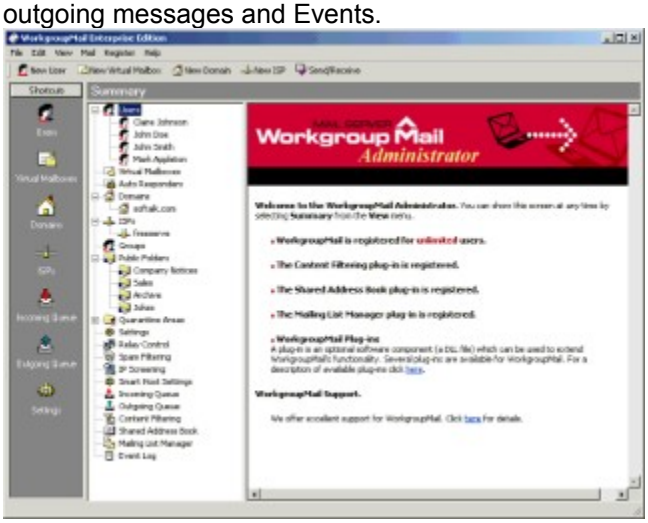
More:

Starting The WorkgroupMail Administrator

Starting The WorkgroupMail Administrator

To start the WorkgroupMail administrator select **WorkgroupMail Administrator** from the **Programs / WorkgroupMail** menu in the **Start** menu.

The WorkgroupMail administrator will appear showing a list of configurable items in the left-hand window. This includes entries for users, virtual mailboxes, auto responders, domains, ISPs, incoming messages, outgoing messages and Events.



Users

A User represents a person in your organization who will be sending and receiving e-mail using WorkgroupMail. A user can receive e-mail destined for one or more mailboxes. You need to add a user for each local POP3 or IMAP account that you wish to make available in your organization.

More:

Adding a New User Manually

Adding Users From the Active Directory

Associating a Manually Added User with the Active Directory

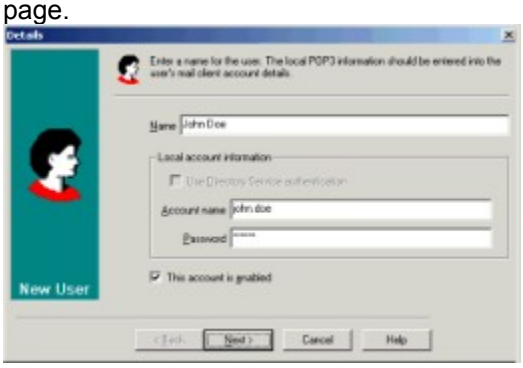
Editing a User's Properties

Deleting a User

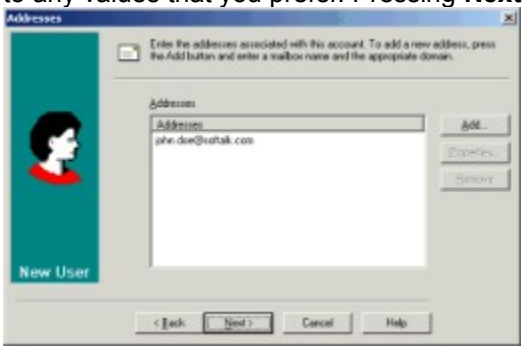
Marking a User as Away from the Office

Adding a New User Manually

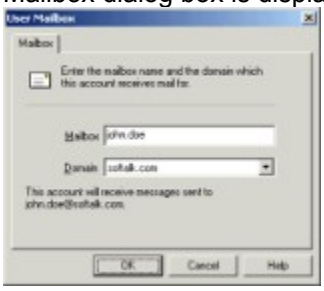
To add a new user, press the New User toolbar button . A New User wizard is displayed showing the Details page.



Enter the full name of the user into the Name field. An auto-generated account name will appear in the **Account Name** field in the Local account information section. The contents of the account name and password fields represent the information that must be entered into a mail client in order to retrieve this user's e-mail from WorkgroupMail into the mail client. You may change the account name and password to any values that you prefer. Pressing **Next** will display the Addresses page.



This page lets you add one or more mailboxes that will be associated with this user. Adding a mailbox defines an additional e-mail address for the user. To add a mailbox, press the **Add…** button. The User Mailbox dialog box is displayed.

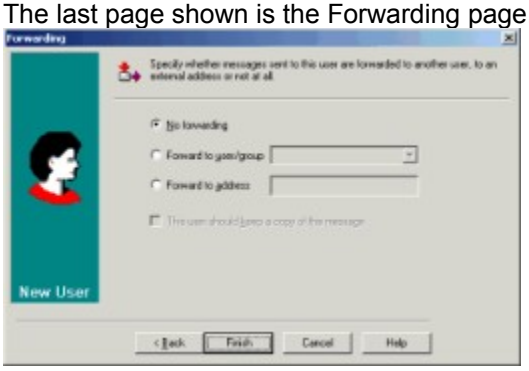


Enter the name of the mailbox into the **Mailbox** field and select the domain to which this mailbox should be added. This action will effectively define an e-mail address for the user. This is displayed at the bottom of the dialog box.

Note: Each mailbox that you add must be unique across the domain. If you try to add a mailbox with the same name as an existing mailbox at that domain then WorkgroupMail will warn you and prevent you from completing the action.

Press **OK** to save your changes and press **Next** on the Addresses page to continue.

The last page shown is the Forwarding page.

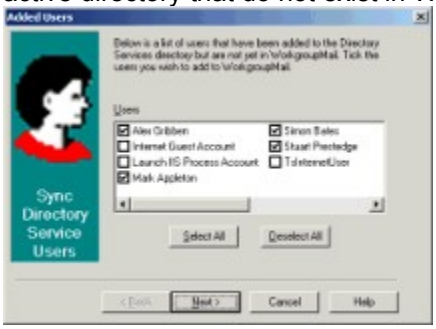


This page lets you specify whether or not e-mail arriving for this user will be forwarded or copied to another user, group or e-mail address. If mail arriving for this user is to be forwarded to another user then select the **Forward to user/group** radio button and select the appropriate user or group. If mail arriving for this user is to be forwarded to an external e-mail address then select the **Forward to address** radio button and enter the e-mail address into the adjacent field. If you have specified to forward this user's e-mail but you also want this user to receive a copy of each message then tick the **This user should keep a copy of the message** tick box.

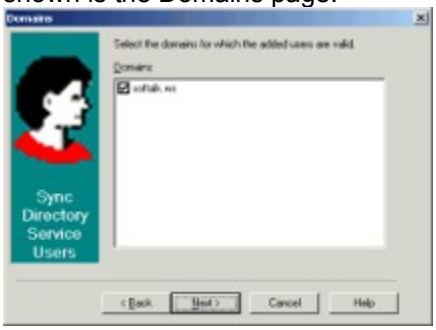
Pressing **Finish** will save the details that you entered for this user and will add the user to the list of users in the WorkgroupMail Administrator.

Adding Users From the Active Directory

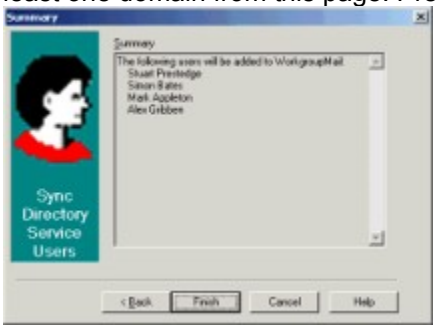
If you have lots of users in your organization, you may find it easier to add users directly from the active directory. Adding users in this way ensures that the users local account name and password is the same as their user name and password in the active directory. To import users from the active directory, select **Edit | Synchronize Directory Service Users**. After a short while, the Sync Directory Service Users property sheet is displayed showing the Added Users page. This page shows a list of users found in the active directory that do not exist in WorkgroupMail.



Select the users that you wish to import. You can use the **Select All** button to select all the listed users. When you have selected the relevant users from the list, press **Next**. The next page shown is the Deleted Users page. This page shows a list of users in WorkgroupMail that were previously imported from the active directory, which no longer exist in the active directory. Selecting any users in this page will ensure that the selected users are removed from WorkgroupMail. Press **Next** to continue. The next page shown is the Domains page.



This page lets you choose which domain(s) the selected users should be added to. You must select at least one domain from this page. Press **Next** to continue. The last page shown is the Summary page.



This page shows a list of users that will be added to WorkgroupMail and a list of users that will be deleted from WorkgroupMail. To perform the action, press **Finish**.

All imported users will be configured to use their Windows credentials as their local account name and password. The local part of their e-mail address will be their Windows user name. For example, if John Smith's Windows user name was **jsmith** and he was added to the **mycompany.com** domain, his e-mail address will be **jsmith@mycompany.com**.

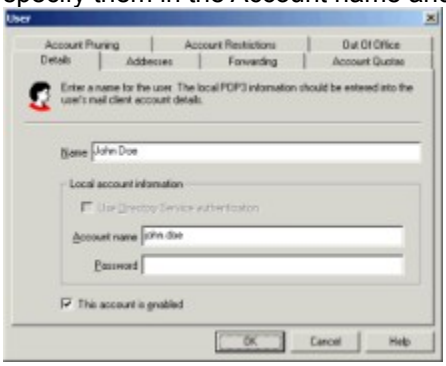
Associating a Manually Added User with the Active Directory

If you have manually added a user that you wish to associate with a user in the active directory, you can do this by right clicking on the user in the left-hand window of the WorkgroupMail Administrator and selecting **Associate with Directory Service User…**. Doing this will display the Select Directory Service User dialog box is displayed.

This dialog box lets you select a user from the drop down list of users available in the active directory. Select the appropriate user to link to and press **OK**. From now on the WorkgroupMail user will be associated with the equivalent user in the active directory.

Editing a User's Properties

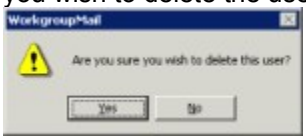
You can change the properties of an existing user by double clicking on the user with the left mouse button. When you do this the User property sheet is displayed, showing the Details page. All the pages in the original New User wizard are available in the property sheet in addition to some advanced pages; Quotas page, Account Pruning page and the Restrictions page, which are discussed later in this section. If the user was imported from the active directory (as explained above), then the **Use Directory Service authentication** tick box will be enabled (and ticked by default), allowing you to specify that this user will get the local account name and password credentials from the active directory, rather than manually specify them in the Account name and Password fields.



Press **OK** to save your changes.

Deleting a User

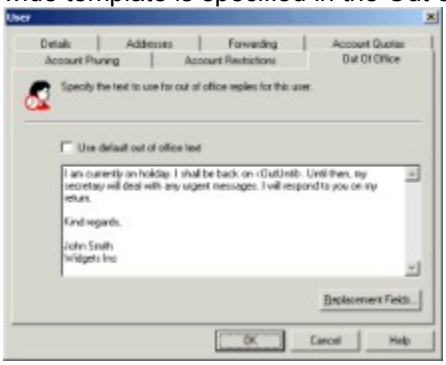
You can delete a user by selecting the user and pressing the **Delete** key. You will be asked to confirm that you wish to delete the user.



Press **OK** to confirm the deletion. It is possible to delete several users at once by selecting the Users entry in the left-hand list of the WorkgroupMail administrator and then selecting the relevant users in the right-hand list and pressing the **Delete** key or by right clicking and selecting **Delete** from the context sensitive menu.

Marking a User as Away from the Office

Quite often, different users may be away from the office, for example, on business trips or on holiday. During these times, they may want to convey the fact that they are away to anyone that sends them email. WorkgroupMail lets either an administrator or the user specify whether they are out of the office and when they are expected back. Once out of the office, any mail arriving for this user will automatically be replied to by WorkgroupMail using an organization-wide template, or using a template specific to that particular user, notifying the sender that the intended recipient is away from the office. The organization-wide template is specified in the Out of Office page of the Settings property sheet.



It is possible to provide a specific response for a particular user by un-ticking the **Use default out of office text** in the Out of Office page of the User property sheet and specifying an appropriate message in the window below.

As an administrator, you can mark any user as being away from the office by right-clicking on the appropriate user and selecting **Away From Office** in the context menu and selecting one of the sub-menu items: **Until Tomorrow**, **Until Next Monday** or **Until…**. If you select **Until…**, an Away From Office Until dialog box is shown, letting you choose the exact date when the user will be back in the office. When the date comes, WorkgroupMail will automatically mark that user as back in the office.

A user can mark themselves as *away from the office* using WebMail. This is discussed in the section on WebMail. For more information, see Overview.

Virtual Mailboxes

A virtual mailbox is similar to a user except mail sent to a virtual mailbox may not be downloaded directly by a local POP3 mail client. Instead, e-mail received by a virtual mailbox must be forwarded on to a WorkgroupMail user.

Unlike a user, you do not need to purchase a license for each virtual mailbox. This means that you can set up an unlimited number of virtual mailboxes for free.

Since virtual mailboxes may be forwarded on to a group, a virtual mailbox is an effective way of distributing messages sent to certain addresses to an entire list of local users.

Virtual mailboxes are typically used if you wish to present an e-mail address to the outside world, such as sales@ourcompany.com or enquiries@ourcompany.com. E-mail then sent to such addresses is not tied to a specific user or group of users. You can change, at any time, the receiver of messages sent to these addresses without changing e-mail address that the sender must send to.

Note: Adding a virtual mailbox is not the only way to enable a user to receive mail sent to more than one defined e-mail address. You can also simply add e-mail addresses to the user, as described in the previous section. However, for e-mail addresses where you will sometimes want to change the associated receiver, it is more convenient to use virtual mailboxes for this task.

More:

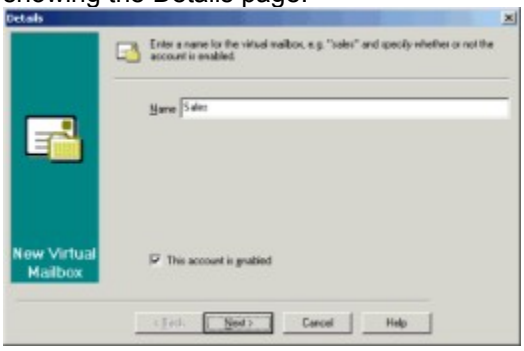
Adding a New Virtual Mailbox

Editing a Virtual Mailbox's Properties

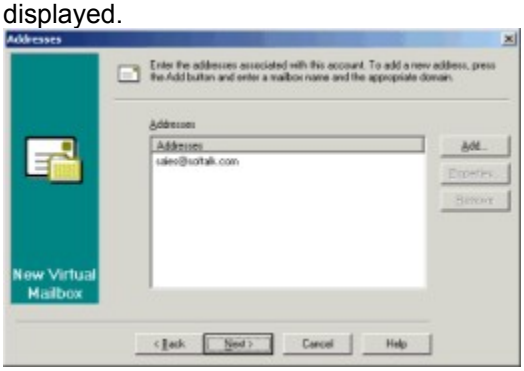
Deleting a Virtual Mailbox

Adding a New Virtual Mailbox

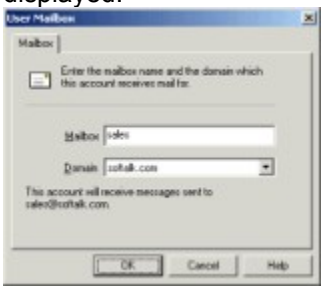
To add a virtual mailbox, press the  toolbar button. A New Virtual Mailbox wizard is displayed, showing the Details page.



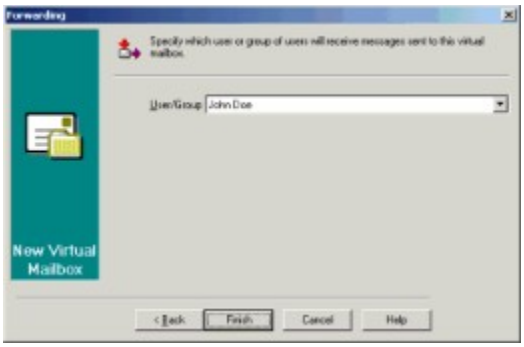
Enter a descriptive name for the virtual mailbox and press **Next** to continue. The Addresses page is displayed.



This page shows the list of e-mail addresses associated with this virtual mailbox. You can define a new e-mail address for the virtual mailbox by pressing the **Add…** button. The User Mailbox dialog box is displayed.



Enter the name of the mailbox into the **Mailbox** field and select the domain to which this mailbox should be added. This action will define an e-mail address for the virtual mailbox. This is displayed at the bottom of the dialog box.Press **OK** to save your changes and press **Next** on the Addresses page to continue. The last page shown is the Forwarding page.

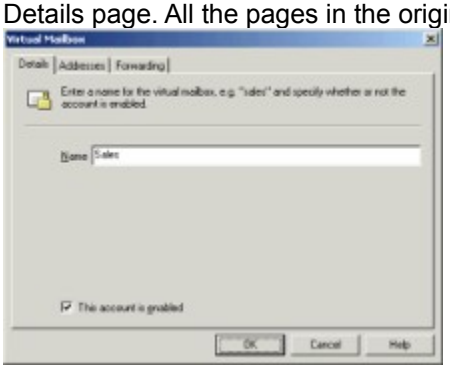


This page lets you specify the user or group that will receive messages sent to this virtual mailbox. Select the appropriate user from the drop down list. If you wish to forward to more than one person then select a group from this list.

Pressing **Finish** will save the details that you entered for this virtual mailbox and will add the virtual mailbox to the list of virtual mailboxes in the WorkgroupMail Administrator.

Editing a Virtual Mailbox's Properties

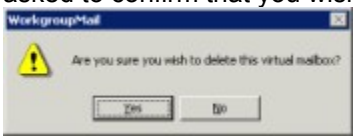
You can change the properties of an existing virtual mailbox by double clicking on the virtual mailbox with the left mouse button. When you do this the Virtual mailbox property sheet is displayed, showing the Details page. All the pages in the original New Virtual mailbox wizard are available in the property sheet.



Press **OK** to save your changes.

Deleting a Virtual Mailbox

You can delete a virtual mailbox by selecting the virtual mailbox and pressing the **Delete** key. You will be asked to confirm that you wish to delete the virtual mailbox.



Press **OK** to confirm the deletion.

Auto Responders

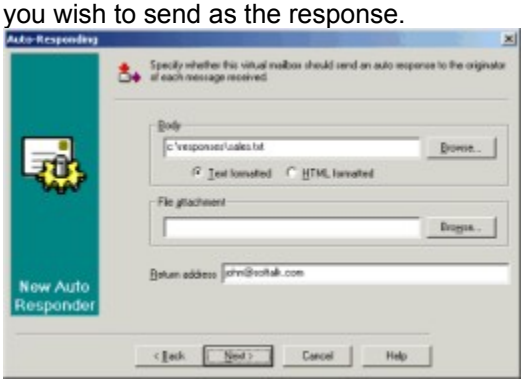
An auto responder is similar to a virtual mailbox except it may be used to send an auto-generated response to the sender. This is very useful, for example, if you wish to provide an instant response every time someone e-mails your enquiries@company.com address, confirming that their message has been received and that their enquiry will be dealt with shortly.

More:

Adding an Auto Responder

Adding an Auto Responder

Adding an auto responder is similar to adding a virtual mailbox. The first two pages are identical. In the auto responder wizard, the page shown after the Address page is the Auto Responding page. This page lets you specify a filename of a file which contains the text or html formatted body of the message that you wish to send as the response.

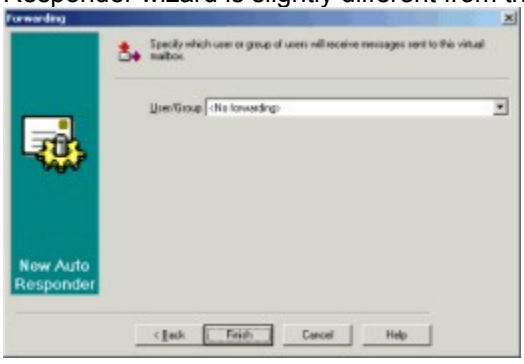


Use the **Browse…** button to select the appropriate file. Select either the **Text formatted** or **HTML formatted** radio button as appropriate.

If the response is to contain a file attachment, then enter the appropriate filename into the File attachment field.

The **Return address** field lets you specify an e-mail address that people should use if they reply to a an auto-response message. If you leave this field blank, the return address will default to the e-mail address of the virtual mailbox. It is always a good idea to change this to an alternative address just in case the e-mail address that the responder sends to is mal-formed. In such an instance, the message may be bounced back to the return address. If the return address is the same as the responder's e-mail address, it is likely that an undesirable response/bounce loop will occur.

Press **Next** to continue. The next page shown is the Forwarding page. The Forwarding page in the Auto Responder wizard is slightly different from the same page in the Virtual Mailbox wizard.



In the User/Group drop down list you have the choice of not forwarding messages received by the auto responder on to a user.

Groups

A group is essentially a collection of users. Groups are useful as a means of forwarding e-mail to more that one person. For example, if you have a virtual mailbox set up to collect mail addressed to sales@mycompany.com and you want to have several people receive this e-mail then the virtual mailbox can be configured to forward the messages to a group of people rather than a single user.

More:

Adding a New Group

Deleting a Group

Adding a New Group

To add a new group, right click on the Groups entry in the left-hand list and select **New Group…** . A Group property sheet will appear.

Enter the name of the group into the *Name* field and then select a user from the **User** drop down list, whom you wish to add to the group, and press the **Add** button. Repeat this procedure for all the other users who should belong to the group and press the **Finish** button to save your changes.

Deleting a Group

You can delete a group by selecting the group and pressing the **Delete** key. You will be asked to confirm that you wish to delete the group.

Press **OK** to confirm the deletion.

Domains

When you open the WorkgroupMail administrator for the first time, after you have run the Setup program, you will notice that there is a sub-entry under the Domains entry. This was added by the Setup program and was extracted from the e-mail address that you supplied in the Setup wizard. For example, if you specified your e-mail address as **john.doe@abcinc.com** then **abcinc.com** will be listed as a domain. In most cases, your organization will own the domain **abcinc.com** and any messages sent to ***anything*@abcinc.com** will be received by WorkgroupMail.

In some cases, particularly true of smaller organizations, the domain may be *borrowed* from your ISP. So, for example, if your e-mail address is **john.doe@myisp.com**, then your domain will be **myisp.com**. In this case, you may have a certain number of e-mail addresses rented from your ISP, e.g. john.doe@myisp.com, mike.smith@myisp.com etc.

Other smaller organizations may *borrow* an entire sub domain from their ISP, for example, **john.doe@mycompany.myisp.com**. In this case, the domain will be listed in WorkgroupMail as **mycompany.myisp.com**.

WorkgroupMail can cater for all these scenarios and can provide full support for multiple domains. For example, if you are an ISP that manages e-mail on behalf of several companies, you can add one domain for each of the companies.

More:

Adding a New Domain

Editing a Domain's Properties

Deleting a Domain

Adding a New Domain

The Setup program automatically adds a domain for you and if you only have a single domain, there should be no reason to add further domains. However, if you do own more than one domain or if you manage e-mail on behalf of other companies or even if you collect mail from several ISPs, each with different domain addresses, then you will need to add further domains.

To add a new domain, press the New Domain toolbar button. The Domain property sheet is displayed showing the Address page.

Enter your domain into the **Domain name** field and press the **Add** button to add the domain to the Domain names list. Press **Next** to show the Mailboxes page. From here, you can add any number of mailboxes and link them to the appropriate user. This is similar to the process for adding a new address for a user and achieves the same result. However, using this interface, you can easily add mailboxes for a number of users.

To add a new mailbox for this domain, press the **Add…** button. The Mailbox dialog box is displayed.

Enter a mailbox name that is unique for this domain and select the user who should receive messages sent to this mailbox from the **Account** drop down list of users. Press **OK** to save your changes. Press **Next** in the Mailboxes page to continue.

The next page shown is the Unknown Recipients page. This page lets you define what happens when a message is received, which is addressed to an e-mail address that has not been defined in WorkgroupMail.

Relay the message

If you tick the **Relay the message if the recipient is found in the routing table** then provided that the relevant recipient address is found in the routing table, the message will be relayed to the server specified in the routing table against the corresponding recipient e-mail address. You can show the routing table to add static routes by selecting the Routing page in the Settings property sheet. This functionality is useful if you are configuring WorkgroupMail to act as a hub for other mail servers that share the same domain. For example, if your organization is split over several physical sites, you might configure WorkgroupMail at one of the sites to receive mail for all users in the organization. Then, by adding static routes for e-mail addresses that correspond to users at the other sites, you can get WorkgroupMail to relay the mail for those users directly to the other WorkgroupMail servers at the other sites.

Reject/delete the message

If you tick the **Reject/delete the message** tick box, WorkgroupMail will refuse to accept any messages sent to e-mail addresses that are not specifically defined at this domain. If the message is received via SMTP (in the case of a standalone configuration), WorkgroupMail will reject the message. The sender of the message will ultimately be notified that the message has been rejected. If the message is received via POP3, WorkgroupMail will delete the message. No warning message will be sent to the sender.

Forward the message

If the message is not rejected and is not relayed then it must be delivered to one of the users. You can

select which user, virtual mailbox or auto-responder will be the recipient of messages sent to unknown mailboxes at this domain. If you select **[Default]** then WorkgroupMail will send such messages to the user that is selected as the recipient of unknown mail across all domains. This user is specified in the Unknown Recipients page of the Settings property sheet.

Pressing the **Next** button will display the Sending page. This page lets you choose whether messages sent from this domain will be sent directly or through your ISPs SMTP server.

If the **Direct** radio button is selected then WorkgroupMail will be operating as a *smart-host server*. In this mode of operation, any message sent from ***someone*@thisdomain.com** will be sent directly to the appropriate destination SMTP server that is responsible for accepting mail for the recipient of the message. WorkgroupMail works out the appropriate SMTP server to connect to by performing an MX lookup using a DNS server. In order for WorkgroupMail to successfully lookup MX records, you must supply a primary (and optionally a secondary) DNS server in the Servers page of the Smart-Host Settings property sheet. Sending directly is more efficient since WorkgroupMail can send several messages concurrently by communicating with several servers at the same time. However, if you dial-up in order to connect to the Internet, this mode of operation is not recommended due to the frequency with which connections are made to destination servers, particularly since WorkgroupMail will frequently retry sending messages that have previously failed.

If you select the **Via** ISP radio button, you must select which ISP's SMTP server to use as the relay server. The drop down list will contain any ISPs that have been defined.

To complete the addition of the domain, press **Finish**.

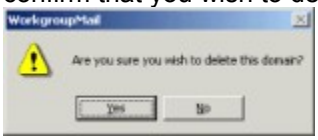
Editing a Domain's Properties

You can change the properties of an existing domain by double clicking on the domain with the left mouse button. When you do this the Domain property sheet is displayed, showing the Address page.

All the pages in the original Domain wizard are available in the property sheet.

Deleting a Domain

You can delete a domain by selecting the domain and pressing the **Delete** key. You will be asked to confirm that you wish to delete the domain.



Press **OK** to confirm the deletion. At least one domain must exist in WorkgroupMail. If you try to delete the last domain, WorkgroupMail will warn you and will prevent you from deleting it.

ISPs

If your organization uses one or more ISPs to send and/or receive mail, then WorkgroupMail must know the details of these ISPs, so that it can communicate with the appropriate servers and query the relevant POP3 accounts.

If you specified in the Setup wizard that your organization collected mail from an ISP, then Setup will have already created an appropriate ISP record for you. If you collect messages from other ISPs then you can add further ISP records in order to do this.

More:

Adding a New ISP

Editing an ISP's Properties

Adding a New ISP

You can add a new ISP by pressing the New ISP toolbar button. A New ISP wizard is displayed.

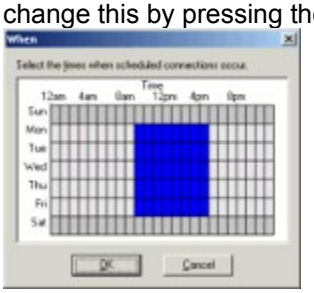
Enter a name for the ISP and press the **Next** button to continue. The next page shown is the Servers page. This page lets you enter information about the mail servers that you use to send and receive e-mail at your ISP.

Enter the name (or dotted IP address) of the appropriate mail servers into the Outgoing mail (SMTP) and Incoming mail (POP3) fields. Press **Next** to continue.

The next page shown is the Connection page. This page lets you specify how and when you connect to this ISP.

If you have a permanent connection to the Internet or you connect to the Internet (for mail) via a proxy server, then select the **Direct connection** radio button. If you need to connect to your ISP through a dial up connection, then select the **Dial up using** radio button and select the appropriate dial up service from the drop down list.

The fields at the bottom of the page let you specify when to connect to the ISP. If you wish to connect when the WorkgroupMail program is started, tick the **On startup after** tick box. If you wish to connect when there are more than *n* messages waiting to be sent, regardless of any other connection options, then tick the **When *n* or more outgoing messages are pending** and enter the appropriate number of messages. If you wish to connect when any message has waited more than a certain number of minutes then tick the **When any message has waited more than *n* minutes** and enter the appropriate number of minutes. If you wish to connect on a regular basis, then tick the **Every *n* minutes** and enter the appropriate interval in minutes. If you tick this tick box then, by default, WorkgroupMail will connect every *n* minutes, only between the hours of 9pm and 5pm, Monday to Friday and not at weekends. You can change this by pressing the **When…** button.



The When dialog box shows a grid with weekdays along one axis and time along the other. The blue areas of the grid represent the times when WorkgroupMail will connect to the ISP every *n* minutes in order to send and receive mail. You can change the color of the grid by clicking the left mouse button and dragging the mouse over the appropriate areas. Press **OK** to save your changes. Press **Next** to continue.

The next page lets you specify information about the POP3 accounts hosted by the ISP.

This page lists all the POP3 accounts that you have already defined. You can add a new POP3 account by pressing the **Add…** button.

If the POP3 account that you wish to add is a single-user POP3 account, that is, it can only be used to download messages for one e-mail address, then select the **Single user** radio button and select, from the drop down list, the user who should receive messages downloaded from this POP3 account.

If the POP3 account that you wish to add is a multi-user POP3 account, that is, it can be used to download messages for several e-mail addresses, then select the **Multiple user** radio button and select, from the drop down list, the domain associated with this POP3 account.

Specify the account name and password required to access this POP3 account in the **Username** and **Password** fields.

To complete the addition of the ISP record, press the **Finish** button.

Editing an ISP's Properties

You can change the properties of an existing ISP by double clicking on the ISP with the left mouse button. When you do this the ISP property sheet is displayed, showing the Details page. Unlike the New ISP wizard, the ISP property sheet has an Advanced page. This page lets you specify advanced settings for the ISP.

Leaving a copy of messages on the server

If you tick the **Leave a copy of messages on the server**, **WorkgroupMail** will still download only those messages that it has not yet already downloaded but will not delete already downloaded messages from the ISP. One possible use of this is to permit two separate sites to download from the same POP3 account without one deleting messages that the other has not yet read. Each site can use any mail client or mail server that can handle this functionality (for example two WorkgroupMail programs, one at each site). Of course, the messages must be deleted at some point. So WorkgroupMail provides a further option to delete messages from the ISP that are more than a certain number of days old. To enable this, tick the **Delete from your ISPs server all mail that is not downloaded and is more that *n* days old**. The latter option can also be used to delete messages that have not been downloaded due to the Junk Mail filtering functionality. See **Error! Reference source not found.** on page **Error! Bookmark not defined.** for more information. It is recommended that this option is used with caution, particularly if there will be many messages remaining on the server for a period of time. In order for WorkgroupMail to determine if it has previously downloaded a particular message, it must download the header of the message. Doing this can be inefficient if there are many messages that WorkgroupMail must sift through in order to find messages that have not yet been downloaded.

Changing the server ports

If your ISP has its POP3 and SMTP servers operating on ports other than 110 and 25, respectively, then you can modify the port numbers from this page to match. Note, it is very unlikely that your ISP will use different port numbers so only change these values if you are an experienced user and are aware of the consequences of doing so

Authenticating with your ISPs SMTP server

Some SMTP servers require authentication. This is particularly the case for organizations that connect to their ISPs servers via DSL rather than dial-up connection (where the latter connection method implicitly authenticates you). ISPs use one of several authentication methods. WorkgroupMail supports the LOGON authentication method.

If your ISP provides you with a username and password to access it's SMTP server then you should tick the **Server requires SMTP authentication** and press the Settings button.

Enter the appropriate login name and password. WorkgroupMail will then use the LOGIN authentication method to authenticate you when accessing the SMTP server for sending mail.

To save the information about the ISP that you have entered, press the **Finish** button. The ISP will be added to the list of ISPs in the WorkgroupMail administrator.

Multi-POP3 account distribution

Some ISPs offer multi-drop (or catch all) POP3 accounts. This provides the ability to collect mail for several users from one POP account. In order for WorkgroupMail to successfully distribute a particular message to the intended local recipient, the ISP must add a header to the message, other than the To: header, indicating the intended recipient address. Most ISPs standardise on providing this information in the Received: header, however some may choose to provide alternative headers, such as Delivered-To: or Apparantly-To. WorkgroupMail lets you specify which header(s) to look for and in which order when determining who a particular message should be forwarded to. You can do this by pressing the **Settings…** button at the bottom of the Advanced page in the ISP property sheet.

The POP3 MIME Headers property sheet is displayed. A list of headers is shown in this page. You can add additional headers by entering the appropriate header into the *Header* field and pressing the **Add…** button. Furthermore, you can use the **Raise** and **Lower** buttons to change the order in which WorkgroupMail will look for headers in determining the appropriate intended recipient.

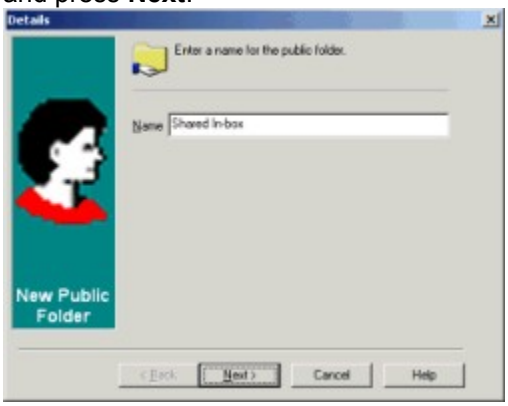
Public Folders

Public folders are message folders which may be accessed by all users of WebMail or IMAP clients that connect to WorkgroupMail. IMAP is discussed in more detail later in this user guide.

Note: IMAP is just another protocol for accessing e-mail. An IMAP client is similar to a POP3 client, except the main difference is that an IMAP client accesses messages that are kept at the server. Unlike POP3, it never downloads messages in order to store them locally. WebMail also relies on messages being stored at the server. Since public folders are also stored at the server, then both WebMail and IMAP clients may access the contents of public folders. For more information on IMAP, see What Is IMAP?.

By defining a public folder, you can effectively create a mail folder to which all users have access. If anyone adds a message to the folder, all other users will be able to access the message. Similarly, if anyone deletes a message from a public folder, the message will no longer be available to any other user.

To create a public folder, right click on the Public Folders entry in the left-hand list of the administrator and select **New Public Folder**. The New Public Folder wizard is displayed. Enter a name for the public folder and press **Next**.



The Rules page is shown next. This page lets you choose what happens to messages held in the public folder. You can choose between keeping the messages indefinitely, moving them to another public folder, after a certain period of time or deleting them after a certain period of time.

Creating a Shared Inbox using Public Folders

One example of using public folders is to set up a shared inbox. Using the content filter, you can add a rule which will copy or move all or certain incoming messages to a specific public folder. Once in a public folder, any user can access these messages in order to read or reply to the message. This is ideal for a

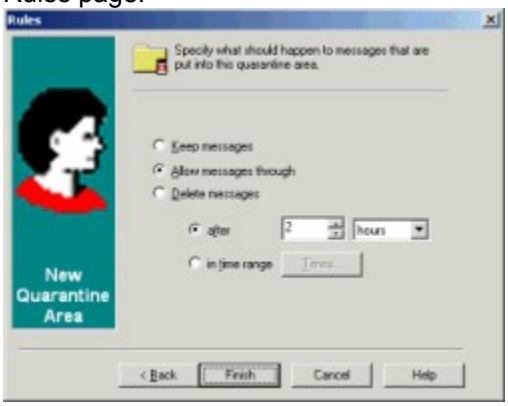
company which employs several staff to provide e-mail support or responses to messages sent.

Quarantine Areas

Quarantine areas are message folders that are available only to an administrator. They are intended as repositories for messages that are identified as not being suitable for delivery to their intended recipient. WorkgroupMail lets you define multiple quarantine areas, thereby making it easy, for example, to keep separate messages that are virus infected from messages that contain unacceptable language. The Virus Protection plug-in and the Content Filtering plug-in can both move messages to selected quarantine areas, as can any custom plug-ins developed by third parties. It is possible to specify rules for quarantine areas so that you can control whether quarantined messages will remain in the quarantine indefinitely, or whether they will be allowed through after a certain amount of time or whether they will be deleted after a certain amount of time.

You can create a new quarantine area by right clicking on the Quarantine Areas entry in the left-hand window of the administrator and selecting **New Quarantine Area…**. The New Quarantine Area wizard is displayed.

Enter a name for the quarantine area and specify an optional description below. Press **Next** to show the Rules page.



This page lets you specify whether messages entering the quarantine will remain there indefinitely until manually removed or whether such messages will be allowed through or deleted after a certain period of time. Press **Finish** to complete the addition of the new quarantine. If you select the **Allow messages through** or **Delete messages** radio button then you can specify the time constraints on this particular option. For example, you may quarantine any messages identified as personal mail and wish to have them kept in the quarantine until after office hours at which point they may be released and sent during a period when the server is not so busy. To achieve this, you can select the **Allow messages through**

radio button and the **in time range** radio button and press the **Times…** button and specify a period during the day when such messages may be released from the quarantine and sent.

Relay Control

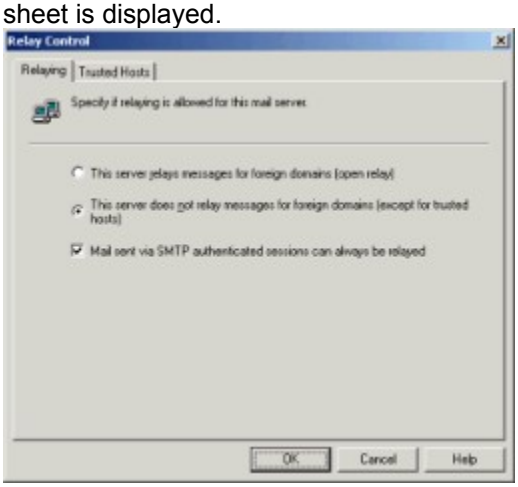
What is Relaying?

The term *relaying* means accepting a message for delivery which is not addressed to a local user. A mail server should be configured to relay messages for either trusted hosts, e.g. all local users, or for authenticated users. Trusted hosts are determined based on their connecting IP address. Authenticated users are identified by a username and password. If a mail server accepts messages for delivery from non-trusted and non-authenticated users then the server is said to be operating as an *open relay*. Mail servers operating as an open relay are an easy target for senders of junk mail and unsolicited commercial email. Certain resources on the Internet try to identify open relay servers and subsequently add them to a black list. The result is that, in an attempt to prevent the ever growing burden of junk mail, messages sent from such servers may be rejected by many mail servers on the Internet. Consequently, if your mail server is permanently connected to the Internet, it is imperative that you correctly configure your relaying settings so as to avoid being added to such black lists.

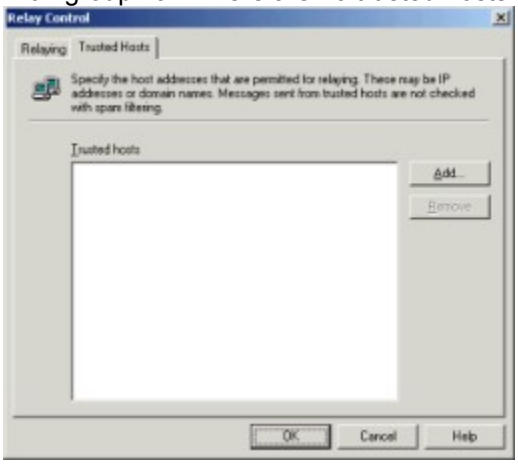
Configuring trusted hosts

If you are operating as an open relay, WorkgroupMail will present you with a warning as soon as you start the WorkgroupMail administrator program.

To resolve this problem, click on the here hotspot as requested. Alternatively, double click on the Relay Settings entry in the left-hand window of the WorkgroupMail administrator. The Relay Control property sheet is displayed.



Select the **This server does not relay messages for foreign domains (except for trusted hosts)** radio button and select the Trusted Hosts page. This page shows a list of the trusted hosts already defined in WorkgroupMail. There are no trusted hosts by default.

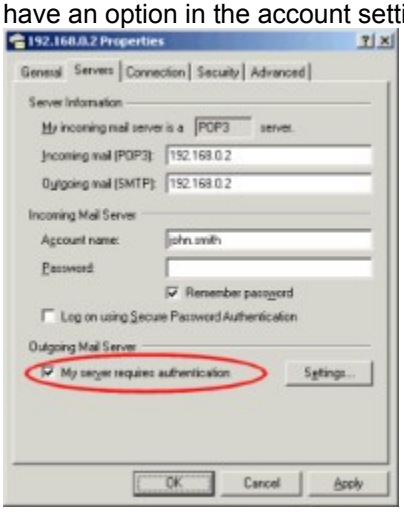


To add a new trusted host, press the **Add** button. The Trusted Hosts dialog box is shown.

This lets you add a single IP address an IP address range or a domain name, that represents one or more trusted hosts.

Configuring client-side authentication

If you prefer to use client-side SMTP authentication rather than or in addition to declaring trusted hosts, you can configure this in each mail client appropriately. For example, both Outlook and Outlook Express have an option in the account settings called **My server requires authentication**.



If you enable this option, you must also specify a username and password to identify the user of the mail client as an authenticated user. The username and password to use for a particular user may be found by selecting the appropriate user from the left-hand window of the WorkgroupMail administrator and looking for the values of *Local account name* and *Local account password*, respectively.

Spam Filtering

Spam filtering of e-mail in WorkgroupMail is provided to enable you to reduce the amount of junk mail that your local users receive. This, in turn, will help reduce the amount of time and resource wasted by your staff in receiving such messages. WorkgroupMail approaches the task of filtering spam in three different ways:

- It can filter based on the IP address of the sender
- It can filter based on the e-mail address of the sender
- It can filter based on the content of the messages

For each incoming message, WorkgroupMail can look at the originator of the message and the content of the message. If the originator's IP address has been black listed or if the message contains known junk mail phrases (defined by you) or if the sender matches one of your black listed junk sender e-mail addresses then WorkgroupMail considers the message to be junk mail and processes it according to the option you have specified. On encountering a junk message, WorkgroupMail can refuse to accept the message (this does not apply when you receive e-mail via POP3), it can delete the message, it can quarantine the message or it can simply mark the message as spam so that it can be processed appropriately by the content filtering component.

More:

Configuring spam filtering

Filtering based on IP address

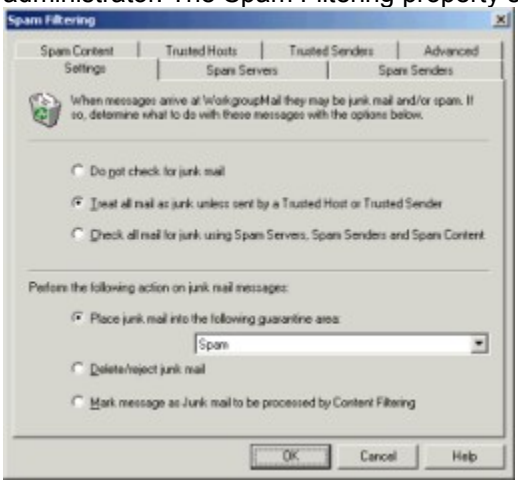
Filtering based on sender e-mail address

Filtering based on content

Trusted Hosts and Senders

Configuring spam filtering

To configure spam filtering, double click on Spam Filtering in the left-hand window in the WorkgroupMail administrator. The Spam Filtering property sheet is displayed, showing the Settings page.



To turn on spam filtering, choose any radio button other than **Do not check for junk mail**. If you select **Treat all mail as junk**, then WorkgroupMail will consider any message that it receives to be a junk message unless the message was sent from a computer with an IP address listed in your Trusted Hosts list or unless the message originated from one of the e-mail addresses listed in your Trusted Senders list. If you select **Check all mail for junk using Spam Servers, Spam Senders and content** radio button, then for each incoming message, WorkgroupMail will contact all of the spam servers listed in the Spam Servers page in order to determine whether the message was sent from an open relay server (a common source of spam mail). It will also look to see if the message originated from one of the e-mail addresses listed in the Spam Senders page and it will look at the content of the message to see if there are any occurrences of the phrases listed in the Spam Content page.

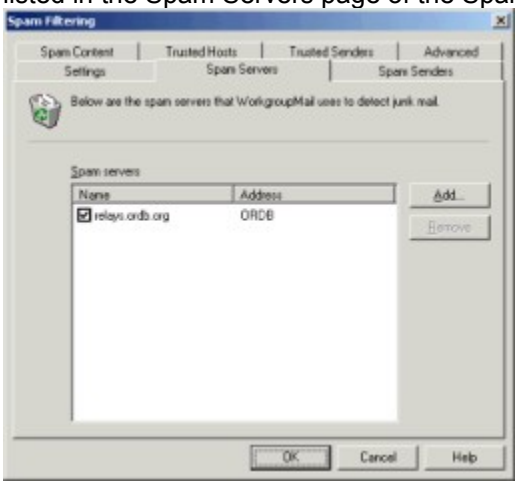
Once WorkgroupMail has deemed a message to be a spam message, you can select one of three actions to be enforced upon the message:

If you select the **Place junk mail into the following quarantine area** radio button then on detection of a spam message, WorkgroupMail will quarantine the message into the chosen quarantine. If you select **Delete/reject junk mail** then on detection of a spam message, WorkgroupMail will either delete the message or reject it depending on whether the message was received by POP3 or SMTP, respectively. If you select the **Add junk mail header to message** radio button, then on detection of a spam message, WorkgroupMail will add a header to the incoming message for later processing by the content filter. This option can make use of the content filter's more powerful action capabilities.

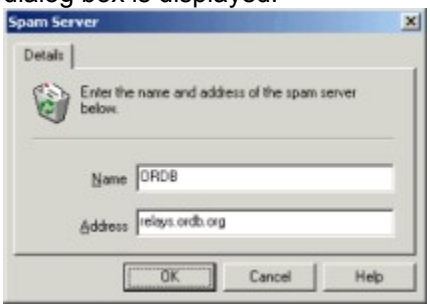
Filtering based on IP address

People who send spam e-mail do so by looking for, and subsequently using, mail servers on the Internet that are not configured to prevent unauthorized users from *relaying* mail. Such mail servers are referred to as *open relays*. There are various servers on the Internet that keep a black list of open relay servers. These servers, which are referred to as *DNSBL servers*, store the IP address of the computer on which the open-relay mail server is running. WorkgroupMail can be configured such that every time a message is received, it checks the originating IP address to see if it is black-listed on any of the RBL servers specified in WorkgroupMail. If it is, WorkgroupMail can reject, quarantine or delete the message, depending on your configuration. Unlike many mail servers, this process works not only when mail is received via SMTP, but also when mail is received via POP3. In the case where a message is received using SMTP, WorkgroupMail looks at the IP address of the connecting SMTP client to determine whether the sender is black listed. In the case where a message is received using POP3, WorkgroupMail will look at the computers that the message passed through before being delivered to WorkgroupMail. If any of the computers are black-listed, the message is considered to be spam.

The list of Spam servers that you are already using to detect messages sent from open relay servers is listed in the Spam Servers page of the Spam Filtering property sheet.



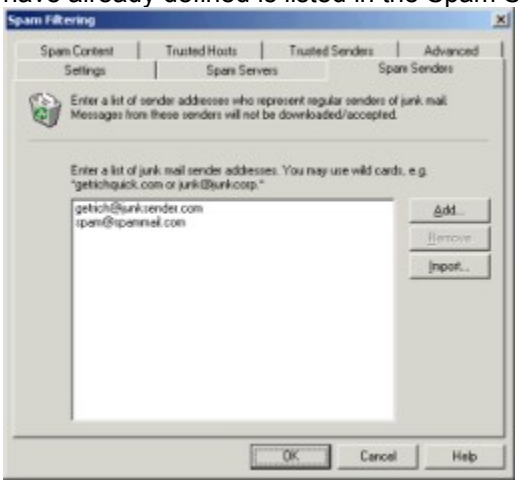
To add a new DNSBL server, select the Spam Servers page and press the **Add…** button. A Spam Server dialog box is displayed.



Enter the name of the spam server that you plan to use. This is for display purposes only. Then enter the address of the spam server, either as a dotted IP address or as a domain address. There are various spam servers available on the Internet. Some charge you to use their service, others are free, such as ORDB. For more information, see http://www.ordb.org. Press **OK** to save your changes.

Filtering based on sender e-mail address

WorkgroupMail also lets you define any number of wildcarded e-mail addresses that you know to be senders of junk mail. On encountering any messages sent from an e-mail address that matches one of the addresses in your spam sender list, WorkgroupMail will mark the message as spam and subsequently reject, delete or quarantine it, according to your preference. The list of spam sender addresses that you have already defined is listed in the Spam Senders page of the Spam Filtering property sheet.

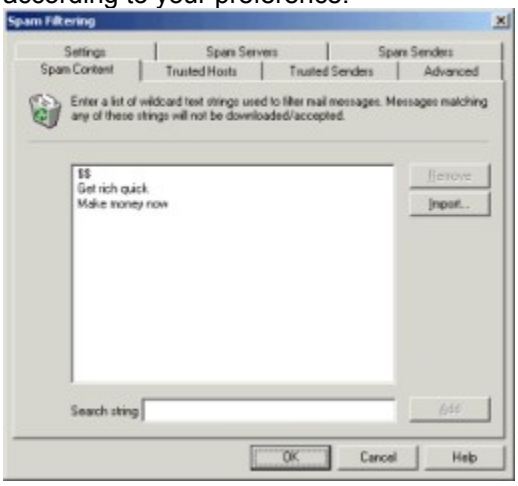


To add a new junk sender, press the **Add…** button. A Enter Junk Mail Address dialog box is displayed.

Enter the full email address or a wildcarded string (representing several email addresses) into the E-mail address field and press **OK**. The new email address will appear in the list. Alternatively, you can import a list of addresses by pressing the **Import…** button and browsing to a file of email addresses. Each email address should be listed on a line of its own.

Filtering based on content

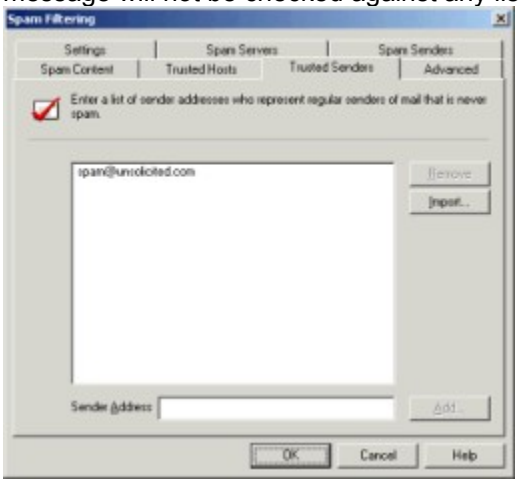
For additional protection, WorkgroupMail lets you specify a set of phrases that are commonly found in junk mail. On encountering any such phrases in the content of any inbound mail messages, WorkgroupMail will mark the message as spam and subsequently reject, delete or quarantine it, according to your preference.



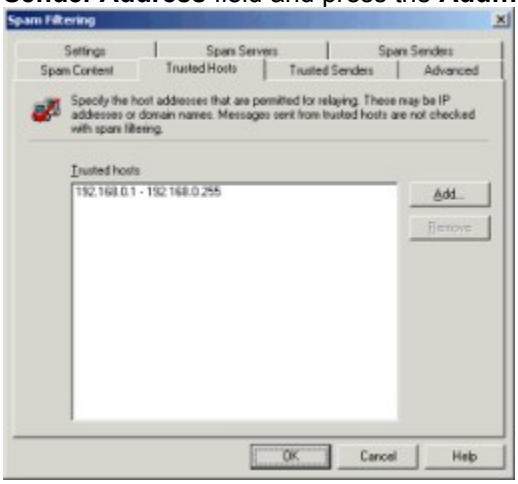
To add a new junk phrase, enter the phrase into the **Search string** field and press the **Add…** button. The phrase will appear in the list above. Alternatively, you can import a list of phrases by pressing the **Import…** button and browsing to a file of phrases. Each phrase should be listed on a line of its own.

Trusted Hosts and Senders

Any message sent from an e-mail address listed in the Trusted Senders page or from any computer whose IP address is listed in the Trusted Hosts page will be automatically deemed as not junk mail. The message will not be checked against any listed spam servers, nor will it be checked for content.



To add an e-mail address into the Trusted Senders page, enter the appropriate e-mail address into the **Sender Address** field and press the **Add…** button. The new sender address will appear in the list above.



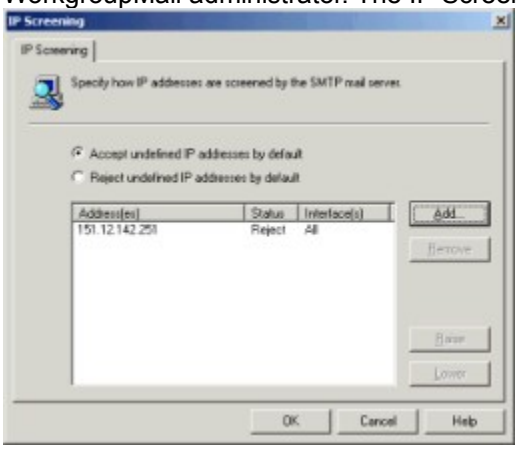
If you know the IP address of a source SMTP server that you trust, then you can enter this IP address or a range of IP addresses into the Trusted Hosts page of the Spam Filtering property sheet. For example, you may choose to enter the entire local area network IP address range to ensure that messages sent from local users are not checked for spam. To add a trusted host, press the **Add…** button in the Trusted Hosts page. The Trusted Hosts dialog box is displayed.

Select whether to specify a single IP address, an IP address range or a domain name and press **OK**.

IP Screening

WorkgroupMail gives you full control over who can and who cannot establish a connection with your mail server in order to send and receive e-mail. By default, WorkgroupMail lets anyone establish a connection to the SMTP, POP3 and IMAP servers. Once a connection has been established, the connecting person (i.e. their mail client) must then go on to authenticate itself in order to send or receive mail. However, if you want to prevent certain persons from connecting at all, you can use the IP screening functionality to restrict which source IP addresses will be accepted or declined when attempting to connect to WorkgroupMail.

To change the IP Screening settings, double click on the IP Screening entry in the left-hand list of the WorkgroupMail administrator. The IP Screening property sheet it displayed.



By default, WorkgroupMail accepts connections from all IP addresses except those specified in the exception list below. You can change this by selecting the **Reject undefined IP addresses by default**. In this case, WorkgroupMail will prevent any computer from connecting except those listed in the exception list. To add an entry to the exception list, press the **Add…** button. An Address(es) dialog box is displayed.

You can specify either a single IP address, an IP address range or a domain name (which resolves to a valid IP address), by selecting the appropriate radio button at the top of the dialog box. You can also specify whether the IP address(es) that you have entered should be rejected or accepted. Finally, you can specify which interfaces this exception applies to. By default, the exception will apply to all interfaces, but if the computer which runs WorkgroupMail has more than one interface (for example a local interface and an Internet interface), then you can select the appropriate interface from the drop down list at the bottom of the dialog box.

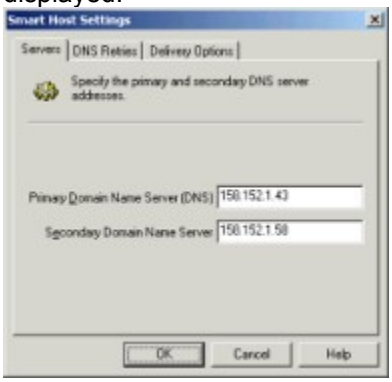
The IP Screening property sheet contains a **Raise** and a **Lower** button. These buttons can be used to

provide exceptions to exceptions. For example, if you wanted to enable all your local users to connect to your server except the range 192.168.0.10 through 192.168.0.20, then you could do this by adding two entries to the list of exceptions:

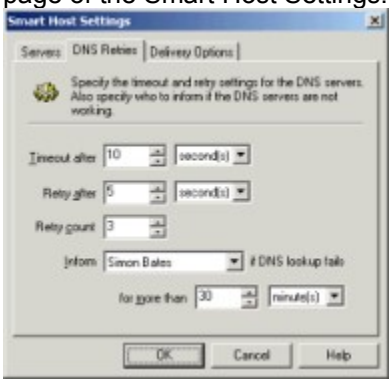
Firstly, you would add the IP address range 192.168.0.1 - 192.168.0.255 and you would select the **Accept** radio button. Then you would add the IP address range 192.168.0.10 - 192.168.0.20 and you would select the **Reject** radio button. You would then select the rejected range and raise them above the accepted range. WorkgroupMail looks from the top of the list and works its way down the list of exceptions. As soon as it finds a match for the connecting IP address, it uses the accept or reject status for that entry.

Smart Host Settings

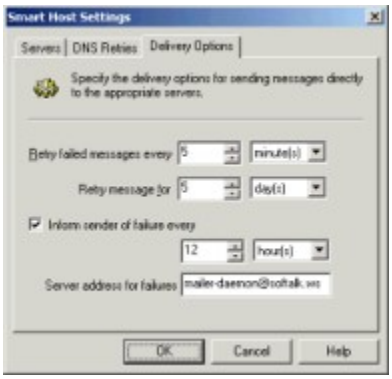
If WorkgroupMail is configured as standalone mail server then it will send e-mail directly to destination SMTP servers rather than using an ISP's SMTP server as a mail relay. In this case, certain additional options may be configured. This can be done by double clicking on the Smart Host Settings entry in the left-hand window of the WorkgroupMail administrator. The Smart Host Settings property sheet is displayed.



When a message is sent, WorkgroupMail contacts the specified primary or secondary DNS servers in order to determine the appropriate SMTP server(s) to connect to in order to send the message. If WorkgroupMail cannot contact either DNS server, after a certain period of time, it will retry for a specified number of times before optionally sending a warning message to the mail administrator or other specified person. The timeout period, retry count and the user that is informed may be specified in the DNS Retries page of the Smart Host Settings.



If WorkgroupMail successfully contacts the DNS server, but finds that there is no server willing to accept messages for a specific recipient domain, WorkgroupMail will immediately return the message back to the sender. If WorkgroupMail successfully finds a destination server but then has problems communicating with that server, then WorkgroupMail will keep the message in the Sent Messages folder and will retry according to the options specified in the Delivery Options page of the Smart Host Settings property sheet.



The settings on this page let you specify how frequently WorkgroupMail will retry sending a message to the destination server in the event of failure. You may also specify the length of time over which WorkgroupMail will continue to try to send the message and the number of hours or days between sending successive warnings to the message originator.

For more information about configuring WorkgroupMail as a standalone mail server see Configuring WorkgroupMail as an Standalone Server.

General Settings

The General Settings property sheet contains a series of pages that let you configure WorkgroupMail for your specific environment. You can show the General Settings property sheet by pressing the  button located in the shortcut bar, or by selecting **Settings…** from the **View** menu.

More:

Multiple-User Account

Unknown Recipients

Logging SMTP/POP3 Communication

Multi-Homed Gateway Computers

Event Log Purging

Routing

Account Quotas

Account Pruning

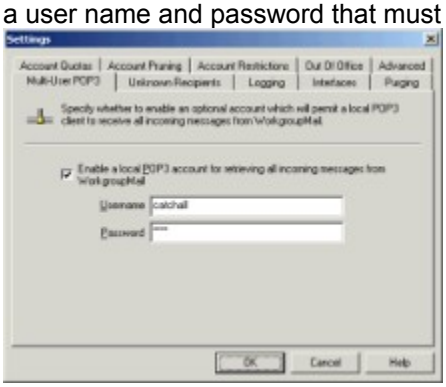
Account Restrictions

Out of Office Settings

Advanced Settings

Multiple-User Account

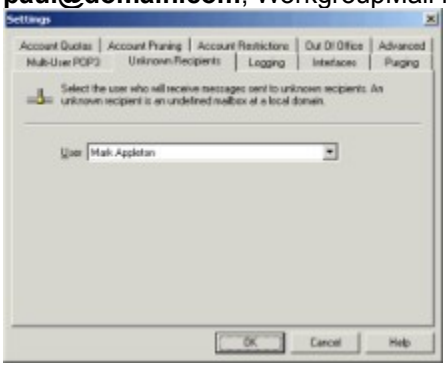
In addition to creating a local POP3 account for each user so that they may download any messages sent to them from the WorkgroupMail message store, it is also possible to enable a local POP3 account that can be used to retrieve all messages for all users from the WorkgroupMail message store. This is sometimes referred to as a catch-all account. You enable the catch-all account by ticking the **Enable a local POP3 account for retrieving all incoming messages from WorkgroupMail** tick box and entering a user name and password that must be used by the mail client in order to access this account.



Note: It is rare that you would want to enable a catch-all account in WorkgroupMail. You may enable such an account if you had another mail server that wanted to receive all the messages received by WorkgroupMail in one go.

Unknown Recipients

If a message arrives that is addressed to a mailbox that does not exist within the relevant domain then the Unknown Recipients page lets you specify what happens to that message. For example, if you have set up WorkgroupMail with the domain **mydomain.com** and you have defined mailboxes **john**, **sue** and **mike**, then WorkgroupMail will know how to deliver messages sent to **john@mydomain.com**, **sue@domain.com** and **mike@domain.com** . However, if a message arrives addressed to **paul@domain.com**, WorkgroupMail needs to know which user to forward this message to.

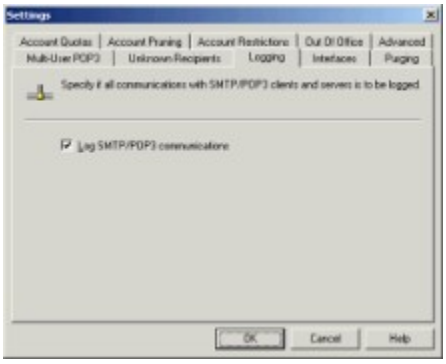


You can specify the user to whom WorkgroupMail will forward all messages sent to unknown recipients by selecting the appropriate user from the drop down list in the Unknown Recipients page. Press **OK** to save your changes.

Note: Individual domains can over-ride this setting if they select anything other than [Default] in the drop down list in the Unknown Recipients page of the Domain property sheet.

Logging SMTP/POP3 Communication

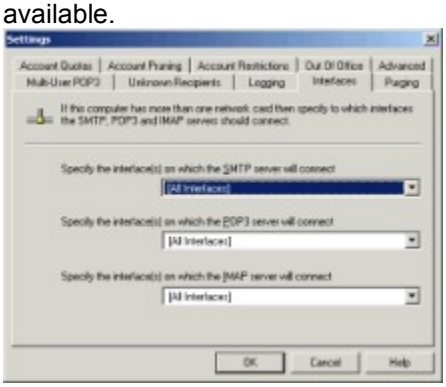
WorkgroupMail lets you keep a complete log of all SMTP and POP3 communication held with both ISP servers and local mail clients. You can enable this log by ticking the **Log SMTP/POP3 communication to a file** tick box.



WorkgroupMail will automatically create log files when WorkgroupMail next communicates with the SMTP and POP3 servers. A new file will be created every day in the **Logs** folder under the Program Files\ WorkgroupMail folder. The name of the file will reflect the current date, eg. **Wmlog_DD_MM_YY.log**.

Multi-Homed Gateway Computers

A multi-homed computer is one that has more than one network card and therefore more than one IP address. By default, WorkgroupMail will listen for IMAP, POP3 and SMTP connections on all IP interfaces. For example, if the gateway computer has an IP address of 45.54.143.245 and 192.168.0.1, either of these IP addresses may be specified as the SMTP, POP3 and IMAP server in the local mail client. However, if one of these interfaces is exposed to the Internet, you may not wish to make the local SMTP and POP3 server available on the external interface, since, unless protected by a firewall, anyone could use the SMTP server to send their own e-mail (if your relay options were not set). For this reason, WorkgroupMail lets you specify the interface(s) on which the local SMTP, POP3 and IMAP servers will be available.



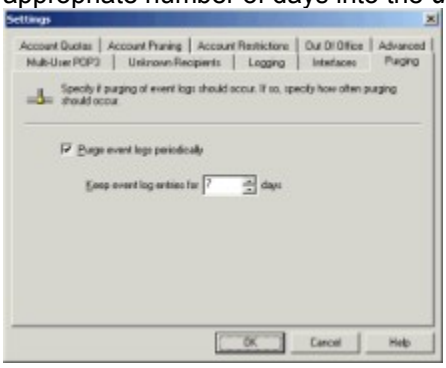
You can specify the interface used for the SMTP server by selecting either **[All Interfaces]** or a specific interface from the top drop down list.

You can specify the interface used for the POP3 server by selecting either **[All Interfaces]** or a specific interface from the middle drop down list.

You can specify the interface used for the IMAP server by selecting either **[All Interfaces]** or a specific interface from the bottom drop down list.

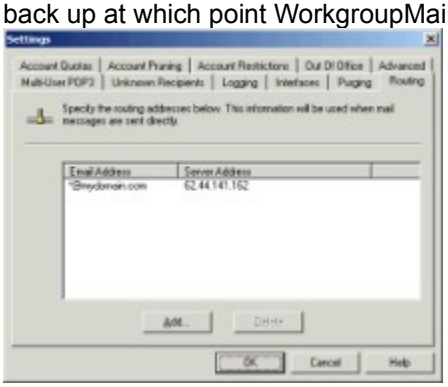
Event Log Purging

You can instruct WorkgroupMail to automatically purge the events in the Event Log that are older than a certain number of days from the Purging page. Tick the **Purge event logs periodically** and enter the appropriate number of days into the *days* fields.



Routing

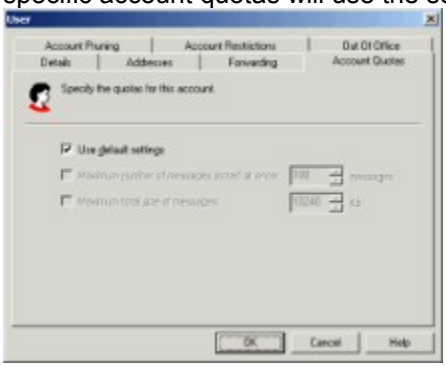
If WorkgroupMail is operating in the standalone configuration (i.e. mail is sent directly rather than via an ISP) then it is possible to specify static routes for sending mail. Normally when a message is sent directly, WorkgroupMail looks at the domain of any recipients and performs several DNS lookups to determine the appropriate mail server(s) to which to send the message. Any static routes that are defined will be used directly rather than using DNS to lookup the appropriate mail server. Being able to enter static routes is useful for several reasons. First of all, if your organization is split over several sites or if you use several mail servers to handle mail for different groups of people in your organization then you can enter static routes to inform WorkgroupMail how to forward on any messages received for specific users. For example, if WorkgroupMail was in charge of receiving mail for the domain **company.com** and **fred.bloggs@company.com** was located at another site and he collected his mail from a separate mail server, then a static route could be entered which could re-direct mail sent to Fred to the appropriate mail server. Another use for static routes is if you want to use WorkgroupMail as a backup server. When your primary server is down, mail will start arriving at WorkgroupMail. A single static route can be added to re-direct mail back to the primary server. The mail will remain in WorkgroupMail until the primary server is back up at which point WorkgroupMail will send the mail on to the primary server.



To add a static route, select the Routing page of the Settings property sheet and press the **Add…** button. The Routing Entry dialog box is displayed. Enter the e-mail address of any recipients that you wish to route. This e-mail address can include wildcards. For example, if you wish to route all mail coming into your domain, then enter ***@yourdomain.com**. In the Server address field, enter the name or IP address of the server to which mail to this address should be routed. Press **OK** to save your changes.

Account Quotas

The Enterprise edition of WorkgroupMail lets you control the maximum number of messages or the maximum size of messages that all users or each specific user is permitted to store. These are called account quotas. You can control the account quotas for a specific user by selecting the Account Quotas page of the relevant User property sheet. You can control the general setting for account quotas by selecting the Account Quotas page of the Settings property sheet. Any user that has not been assigned specific account quotas will use the settings in this page.

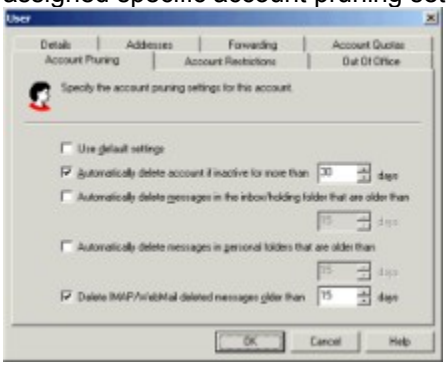


Both pages let you define the maximum number of messages that may be stored at once and the maximum total size of all messages that may be stored. The Account Quotas page in the User property sheet has a **Use default settings** tick box. If this is unticked, the remaining controls in this page will become enabled, letting you specify individual quota settings for this user.

If a message arrives for any user that has exceed their account quota then, provided the message has been received via SMTP (account quotas do not work when receiving mail via POP3), the message will be rejected and the connecting SMTP server will return the message back to the sender informing them that the message was rejected because the user had exceeded their account quota.

Account Pruning

The Enterprise edition of WorkgroupMail can automatically delete certain messages older than a certain number of days and can automatically delete user accounts that have been inactive for more than a certain number of days. This activity is referred to as *account pruning*. Account pruning helps keep control over the amount of disk space used by users' message stores and removes the need for manual maintenance of the data store. You can control the account pruning for a specific user by selecting the Account Pruning page of the relevant User property sheet. You can control the general setting for account pruning by selecting the Account Pruning page of the Settings property sheet. Any user that has not been assigned specific account pruning settings will use the settings in this page.

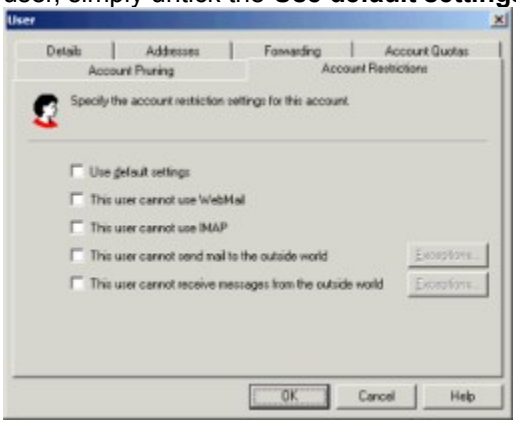


The Account Pruning page lets you choose to automatically delete a user account if it has been inactive for more than a certain number of days. An inactive account is one that has not been logged into from neither POP3 nor IMAP within the specified number of days. To automatically delete inactive user accounts, tick the **Automatically delete account if inactive for more than n days** and specify the appropriate number of days.

Another option lets you choose to automatically delete messages in a user's inbox that are more than a certain number of days old. You can do this by ticking the **Automatically delete messages in the inbox/holding folder than are older than n days** tick box and specify the appropriate number of days.

Account Restrictions

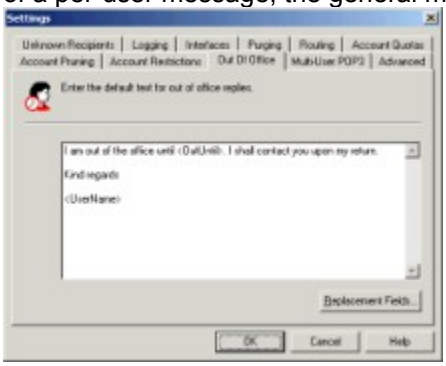
The Enterprise edition of WorkgroupMail makes it possible to restrict certain users from using WebMail or from connecting to the IMAP server or from sending and receiving mail externally. This can be done unilaterally or on a per-user basis. The Account Restrictions page in the Settings property sheet lets you specify, on a global basis, which restrictions apply to all users. The Account Restrictions page in the User property sheet lets you specify which restrictions apply to a specific user. By default the **Use default settings** tick box is selected in the Account Restrictions page for each user. This means that all users use the settings specified in the Settings property sheet. To change the restriction settings for a particular user, simply untick the **Use default settings** tick box and tick the appropriate restrictions.



For example, if you wanted to prevent a certain user from sending mail externally, except to certain recipients, you could do this by ticking the **This user cannot send mail to the outside world** tick box and pressing the **Exceptions** button and specifying a list of the people to whom the user may send mail. If you wanted to prevent the user from receiving mail except from certain senders, you could do this by ticking the **This user cannot receive messages from the outside world** and pressing the **Exceptions** button and specifying a list of the people from whom the user may receive mail.

Out of Office Settings

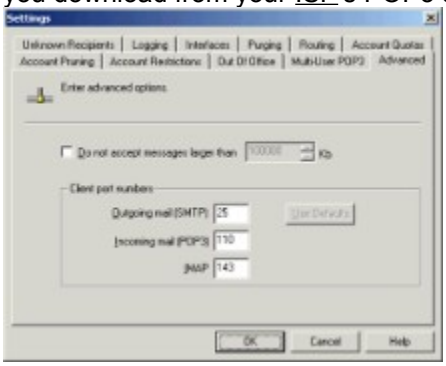
WorkgroupMail makes it easy to specify when a user is out of the office. When this is specified in WorkgroupMail, all mail for the relevant user is automatically responded to, informing the sender that the user is away from the office and informing them when the user will be back. The message that is sent to the sender of each message may be specified on a per-user basis or on a general basis. In the absence of a per-user message, the general message is used to respond to senders.



The Out of Office page of the Settings property sheet lets you specify a message that will be the default message sent back to any person that sends a message to a user that is currently marked as *Out of the Office*. There are two replacement fields that may be used to provide more context to the response message. The first replacement field is **<OutUntil>**, which is replaced by the date when the user is back in the office. The second replacement field is **<UserName>**, which is replaced by the relevant user's full name.

Advanced Settings

WorkgroupMail lets you specify whether an upper limit should be imposed on the size of messages that you download from your ISP's POP3 server. You can do this from within the Advanced page.



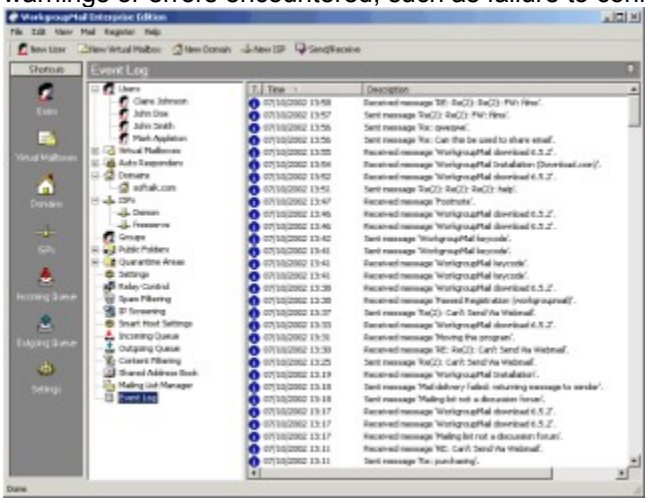
Tick the **Do not download messages larger than** tick box and enter the appropriate value into the edit field.

If you have other mail server software or other similar software running on the same computer as WorkgroupMail, you may need to change the port numbers that WorkgroupMail uses to listen to its clients. If you change the port numbers to anything other than 25, 110, 143 for Outgoing mail, Incoming mail and IMAP, respectively, then the same change must be made in the advanced settings of the client e-mail software.

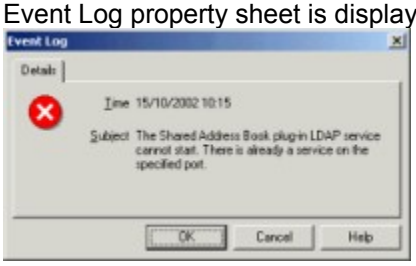
Press **OK** to save your changes.

The Event Log

The event log records any message that is sent and received through WorkgroupMail. It also records any warnings or errors encountered, such as failure to connect to POP3 servers etc.



You can view the event log by clicking on the Event Log entry in the left-hand list of the WorkgroupMail administrator. If you cannot see the full description of the event, double click on the relevant event. An Event Log property sheet is displayed showing the full description of the event.

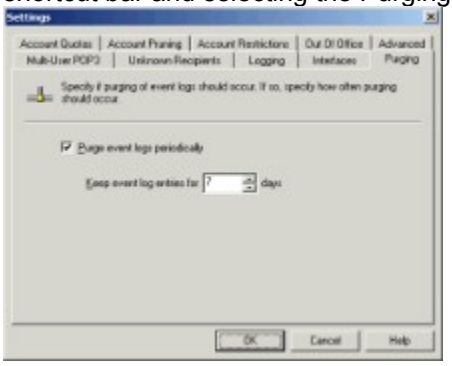


More:

Purging Events

Purging Events

You cannot delete events in the event log. Instead, WorkgroupMail automatically purges any events that are over 7 days old. You can change the maximum age of events by pressing the Settings button in the shortcut bar and selecting the Purging page.



Specify the number of days to keep entries for and press **OK** to save your changes.

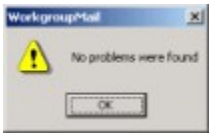
Viewing Sent and Received Messages

The WorkgroupMail Administrator lets you view incoming messages that are soon to be delivered to local recipients and outgoing messages that are soon to be sent on to the Internet for delivery to their intended recipient.

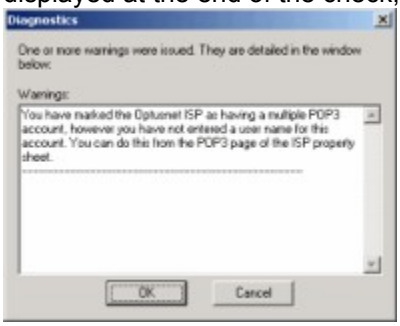
You can do this, simply, by clicking on the **Received Messages** or **Sent Messages** entry as appropriate in the left-hand window of the WorkgroupMail administrator.

Running Diagnostics

WorkgroupMail has a way of determining whether or not WorkgroupMail is correctly set-up. It does not catch all problems but it is a good start. To run the Diagnostic Checker, select **Run Diagnostics** from the **Edit** menu. If there are no problems, a message box is displayed confirming this:



If there are specific problems that need your interaction, the appropriate message boxes will be displayed, prompting you for various inputs. If there are one or more warnings, then a Diagnostics dialog box is displayed at the end of the check, listing all the warnings that were found.



Make any changes to the settings, as appropriate, and then re-run the diagnostics check to ensure that there are no remaining warnings.

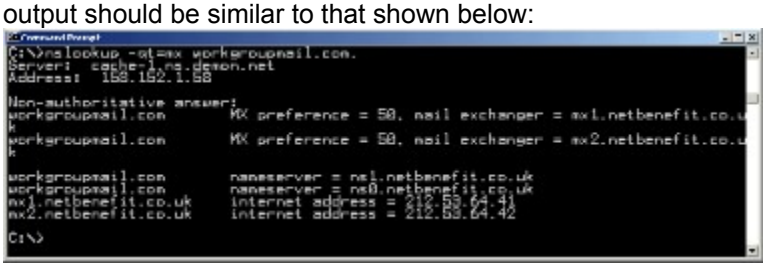
Configuring WorkgroupMail as an Standalone Server

WorkgroupMail can be configured to work either in conjunction with an Internet Service Provider (ISP) or to work independently of an ISP. The latter case configuration is referred to as *standalone* Mail Server configuration.

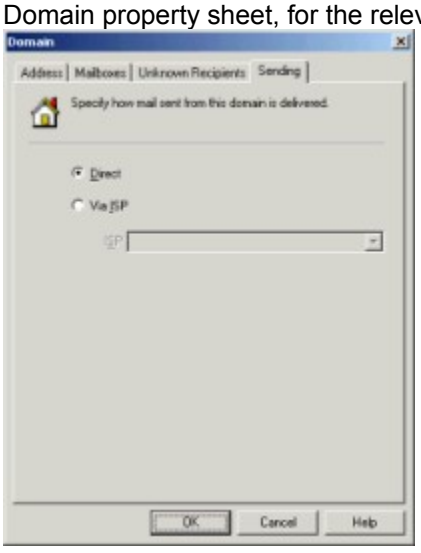
In order to use WorkgroupMail as standalone mail server, you must have a permanent connection to the Internet and a mail exchange record (MX record) must exist which points to the computer running WorkgroupMail. Ask your domain provider to configure the MX record if it is not already pointing to your server. You can tell whether the MX record for your domain points to the correct IP address by typing the following into a command window:

nslookup -qt=mx mydomain.com.

You should replace mydomain.com with your domain name. Note the trailing period is required. The output should be similar to that shown below:

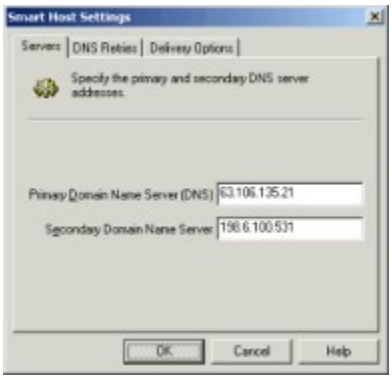


One or more result lines should show an MX preference result with the mail exchanger pointing to the name or IP address of the computer which runs WorkgroupMail. Provided the response is successful, it will be possible to configure WorkgroupMail as a standalone mail server.
The best way to configure WorkgroupMail as standalone Mail Server is to run the installation program and select **Standalone mail server** from the Configuration page of the Setup wizard. To specify advanced settings or to re-configure WorkgroupMail as a standalone mail server after you have installed, open the Domain property sheet, for the relevant domain, and select the Sending page.



Ensure that the **Direct** radio button is selected. Press **OK** to save your changes.

Next, open the Smart Host Settings property sheet by double clicking on the Smart Host Settings entry in the left-hand window of the WorkgroupMail administrator. Select the Servers page.



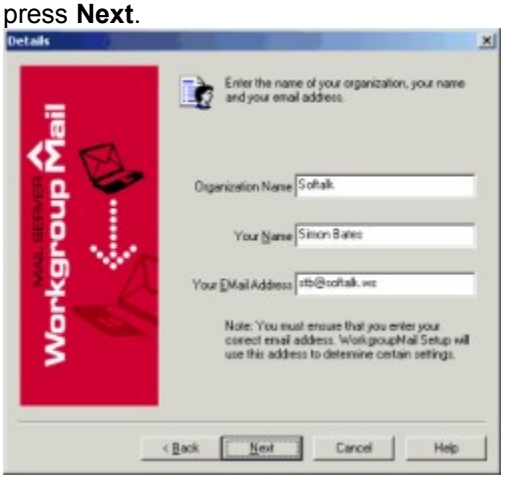
Enter the IP address of your primary and secondary DNS servers. When you press **OK**, WorkgroupMail will now be configured as an enterprise mail server. WorkgroupMail will now receive e-mail sent directly by source SMTP servers and when a user sends a mail message, it will now be routed directly to the recipients' destination SMTP server.

Note that it is still possible to collect mail from an ISP using POP3 when configured as a standalone server. This can be done by configuring another domain to collect mail from an ISP, rather than directly in the Sending page of the appropriate Domain property sheet.

Using WorkgroupMail for Internal Mail Only

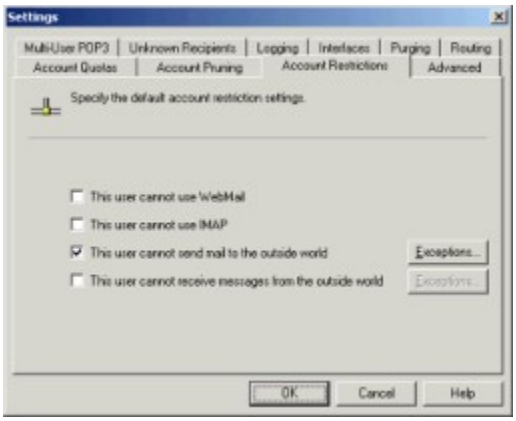
Some organizations don't wish to provide Internet e-mail capabilities to their employees but instead just wish to provide internal e-mail. This is done easily by running the installation wizard and selecting the Internal Mail Only radio button in the Configuration page.

Press **Next**. In the next page, enter your organization's details, your name and your e-mail address and press **Next**.

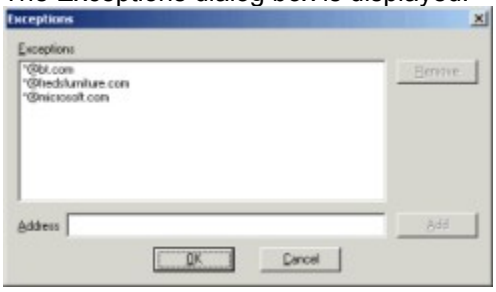


The last page is the Summary page which summarizes the options that you have chosen. Press **Finish** in this page.

When you press **Finish** you will be configured for internal mail for one user. Use the **WorkgroupMail** administrator to add further users. WorkgroupMail Enterprise version gives you the ability to prevent users from sending external messages. To prevent all users from sending Internet mail, press the Settings button and go to the Account Restrictions page.

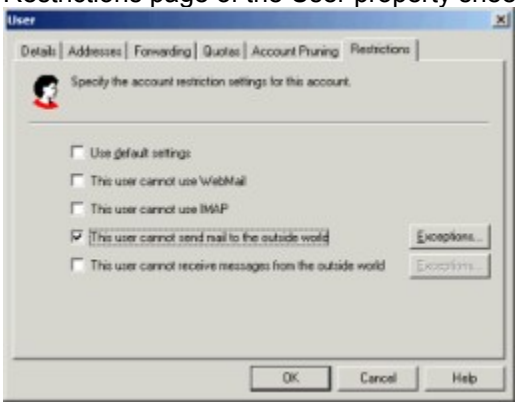


Tick the **This user cannot send mail to the outside world** tick box. If you prevent all users from sending to the Internet except to certain e-mail addresses, then you can press the **Exceptions…** button. The Exceptions dialog box is displayed.



Enter a e-mail address (which may include wildcards) into the Address field and press the **Add** button. WorkgroupMail will then prevent all users from sending messages externally unless the recipients to which a message is sent match those listed in the Exceptions list.

If you want to restrict only certain users from sending mail externally then you can do this from the Restrictions page of the User property sheet (Enterprise edition only).

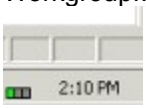


Untick the **Use default settings** tick box and tick the **This user cannot send mail to the outside world** tick box. If there are any exceptions related to this particular user then press the **Exceptions** button and specify them in the same way as described above.

Running WorkgroupMail As a Program

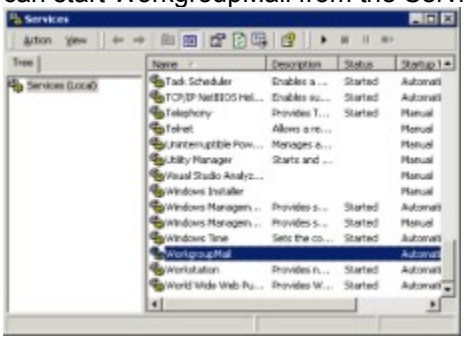
If you installed WorkgroupMail as an executable program, rather than an NT service (does not apply to Windows 95/98/ME users), then you can start WorkgroupMail by selecting **Start WorkgroupMail** from the **Start / Programs / WorkgroupMail** menu in Windows.

An animated icon will appear in the task bar. If the lights are lighting in sequence, this means that WorkgroupMail is operating correctly.



Running WorkgroupMail as an NT Service

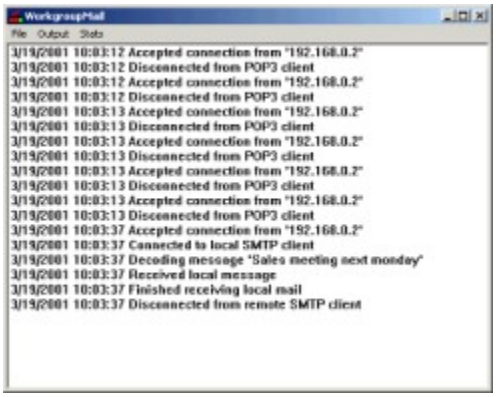
If you installed WorkgroupMail as an NT service (does not apply to Windows 95/98/ME users), then you can start WorkgroupMail from the Services applet in Administrator Tools (or control panel on NT 4.0).



Select WorkgroupMail from the list of services and press the ▶ button in the toolbar. WorkgroupMail will automatically start when the machine is restarted. If you wish to prevent this from happening, double click on the WorkgroupMail entry and select **Manual** from the Startup Type drop down.

Viewing The Output Window

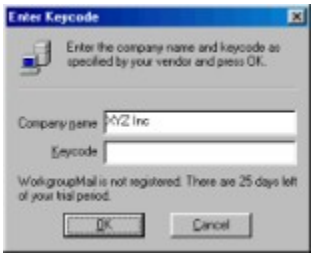
Once WorkgroupMail is started, all activity is written to the Output window. This includes the download and upload progress of large messages. To view the Output window, click on the icon using the right mouse button and select **Show Window** from the context menu. The Output window will be displayed.



Licensing WorkgroupMail

In order to continue using WorkgroupMail beyond the 30-day trial period, you must purchase a keycode. You can do this by visiting **http://www.workgroupmail.com** and pressing the Purchase link. If you purchase on-line, you will be presented with an on-line form to fill out where you can specify how many user licenses you wish to purchase along with your credit card details. Once you have submitted the form and your payment has been accepted, a short while later you will receive an e-mail message specifying your keycode.

You can enter the keycode by selecting Enter Keycode from the Register menu in the WorkgroupMail administrator.



You should enter the company name, exactly as specified in the e-mail message and then enter the keycode, again, exactly as specified in the e-mail message. When you have entered the last character of the keycode, the text at the bottom of the Enter Keycode dialog box will change to signify that the keycode has been accepted and will indicate the number of user licences that you have purchased. Press **OK** to save the changes. From now on when you start the WorkgroupMail program, you will no longer see the warning message, indicating the number of days left in the trial period. You now have a fully licensed version of WorkgroupMail.

Setting up E-mail Clients

WorkgroupMail may be used in conjunction with any POP3 enabled e-mail client. This includes Outlook, Outlook Express and most other Internet enabled mail clients.

Once WorkgroupMail is configured so that it can communicate with the ISP and all the user accounts have been set up in the WorkgroupMail administrator, you are now ready to set up the e-mail clients. All e-mail clients will provide you with a way to specify the SMTP (outgoing mail) and POP3 (incoming mail) servers and to enter an account name and password to gain access to the POP3 account. To configure the e-mail client to work with WorkgroupMail, you should specify the four pieces of information as follows:

SMTP Server (Outgoing mail)	This should be set to the IP address of the computer running WorkgroupMail.
POP3 Server (Incoming mail)	This should also be set to the IP address of the computer running WorkgroupMail.
Account name	If you double click on the user in the WorkgroupMail administrator, whose e-mail client you are configuring, the User property sheet will be displayed. At the bottom of the Details page you will see Local POP3 information. The contents of the Account Name field is what you should enter into the Account Name field of the mail client settings.
Password	If you double click on the user in the WorkgroupMail administrator, whose e-mail client you are configuring, the User property sheet will be displayed. At the bottom of the Details page you will see Local POP3 information. The contents of the Password field is what you should enter into the Password field of the mail client settings.

Once this information has been entered, you should be able to check for new mail in the mail client. You should also be able to compose a message and send it. You will see it appear as a Sent Message in the WorkgroupMail administrator.

More:

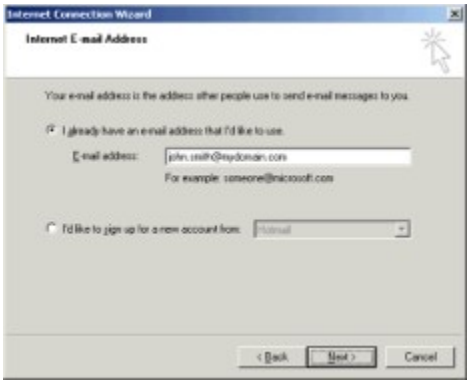
Configuring Outlook Express

Configuring Outlook

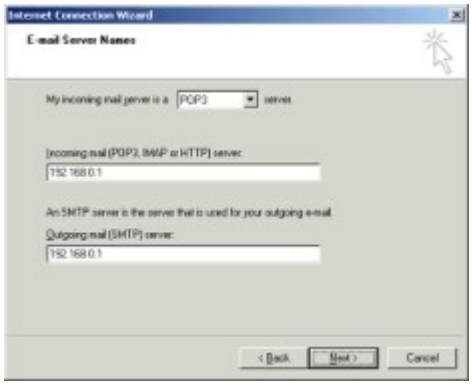
Configuring Outlook Express

To configure Outlook Express, follow this procedure:

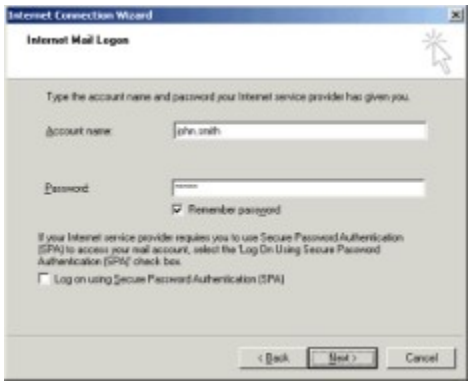
- In Outlook Express, select **Tools | Accounts…** and select the **Mail** tab.
- Delete any mail accounts that exist by selecting them and pressing the **Remove** button.
- Press the **Add -> Mail…** button. The Internet Connection wizard will appear.
- In the Display name field, enter the full name of the person whose mail client you are configuring and press **Next**.
- In the next page, select the **I already have an e-mail address…** radio button and enter the person's e-mail address into the **E-mail address** field.



- Press **Next** to continue. In the **E-mail Server Names** page, specify that the incoming mail server is a **POP3** mail server and enter the IP address of the computer which runs WorkgroupMail into the **Incoming Mail (POP3) Server** and **Outgoing Mail (SMTP) Server** fields.



- Press **Next** to continue. In the **Internet Mail Logon** page, enter the Local POP3 Login Name and Local POP3 Password, for the person whose mail client you are configuring, into the **Account name** and **Password** fields, respectively.

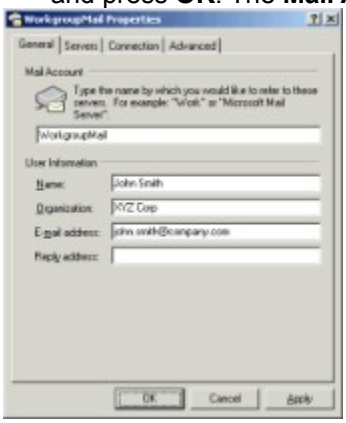


- You can find out the Local POP3 Login Name and Local POP3 Password by selecting the appropriate user in the left-hand list of the WorkgroupMail administrator and viewing the appropriate information in the right-hand window. Press **Next** to continue. This will take you to the **Finish** page. Click on the **Finish** button to complete the configuration. Provided that the WorkgroupMail server program is running, you should now be able to send and receive mail to and from WorkgroupMail.

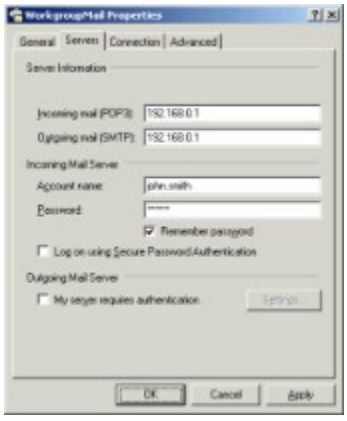
Configuring Outlook

To configure Outlook, follow this procedure:

- In Outlook, select **Tools | Services…** . The Services dialog box is displayed with a list of the currently installed services. Remove any services that begin with **Internet E-mail**. Do this by selecting them and pressing the **Remove** button.
- Press the **Add…** button The **Add Service to Profile** dialog box is displayed. Select **Internet E-mail** and press **OK**. The **Mail Account Properties** sheet is displayed showing the General page.



- Enter the full name of the person whose mail client you are configuring into the **Name** field. Enter the name of the organization on the **Organization** field and enter the person's e-mail address in the **E-mail address** field.

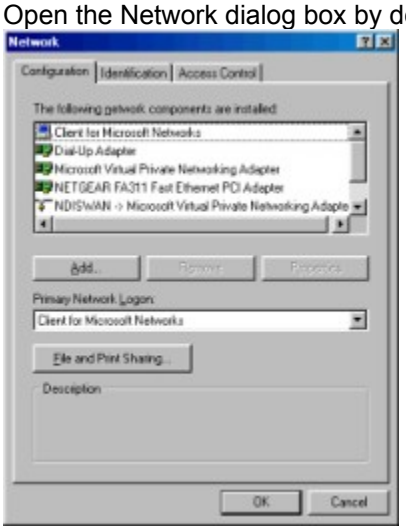


- Select the **Servers** page. Enter the IP address of the computer which runs WorkgroupMail into the **Incoming Mail (POP3)** and **Outgoing Mail (SMTP)** fields.
- Enter the Local POP3 Login Name and Local POP3 Password, for the person whose mail client you are configuring, into the **Account name** and **Password** fields, respectively. You can find out the Local POP3 Login Name and Local POP3 Password by selecting the appropriate user in the left-hand list of the WorkgroupMail administrator and viewing the appropriate information in the right-hand window.
- Select the Connection page. Ensure that the **Connect using my local area network (LAN)** radio button is selected.
- Press **OK** to complete the configuration. Provided that the WorkgroupMail server program is running, you should now be able to send and receive mail to and from WorkgroupMail.

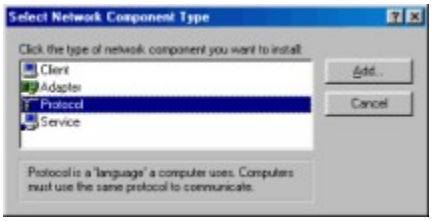
Installing TCP/IP

TCP/IP is the networking protocol used by the Internet and Internet based applications. It comes for free with Windows and it is fairly straight forward to install. POP3 based e-mail clients such as Outlook and Outlook Express use SMTP and POP3 to send and receive e-mail to and from the computer running WorkgroupMail. POP3 and SMTP are TCP/IP-based protocols and so, in order to work, TCP/IP must be installed. Most computers these days usually have TCP/IP installed. If it is not installed on your network, then follow this simple procedure for setting it up:

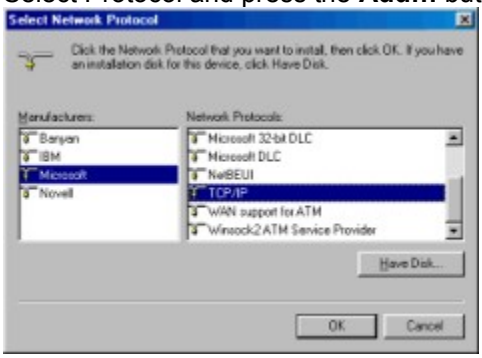
Open the Network dialog box by double clicking on the Network applet in Control Panel.



Click the **Add…** button. The Select Network Component Type dialog box is displayed.



Select Protocol and press the **Add…** button. The Select Network Protocol dialog box is displayed.



In the left-hand window, select **Microsoft**. In the right-hand window, select TCP/IP and press **OK**. When you do this, a TCP/IP entry will appear in the Network dialog box for each network adapter connected to the computer. In most cases, there will just be one entry, corresponding to the network card in the computer. If the computer has a modem, there will also be a TCP/IP entry binding to a dial-up adapter.

Select the TCP/IP entry that is bound to your internal network card and press the **Properties…** button. The TCP/IP Properties dialog box will appear.

In the IP Address page, you must specify an IP address and sub-net mask for the computer. Each computer in your network should have a different IP address to identify itself uniquely. Enter the address 192.168.0.1 for the first computer and then assign 192.168.0.2 to the second computer, 192.168.0.3 to the third computer and so on. All computers should have the same sub-net mask which should be set to 255.255.255.0 .

Press **OK** to save your changes. Windows may ask you to re-boot the computer. When you do this, TCP/IP will be installed on this computer. The same procedure should be performed on each computer in your network, remembering to assigned each computer a different IP address as explained above.

Overview

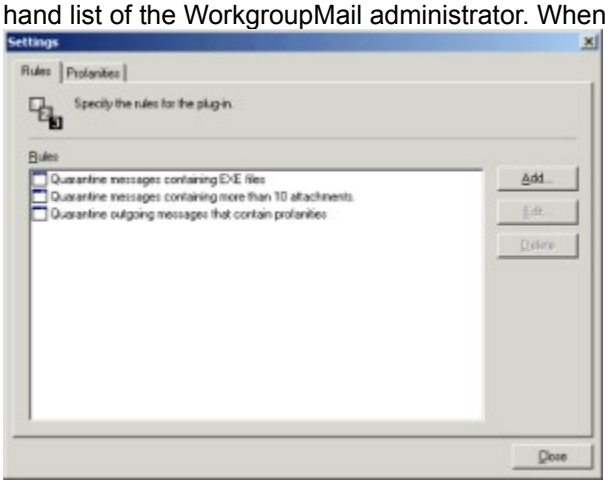
The content filter provides you with a powerful set of functions for content checking of both incoming and outgoing messages. For example, you may wish to check any outgoing messages for viruses or unacceptable language or you may wish to prevent users from sending out company sensitive information or stop certain users from receiving messages containing JPEG or GIF attachments, or automatically add legal and commercial disclaimers to the end of each message. All this and much more is possible using the content filtering component.

WorkgroupMail comes supplied with the content filtering component pre-installed. This feature is included in the Enterprise edition of WorkgroupMail. It is an optional plug-in for the Professional edition of WorkgroupMail and it is not available in the Home edition of WorkgroupMail.

Note: This component will function during the 30-day trial period, but must be purchased separately if used in the Professional edition of WorkgroupMail after the 30-day trial period.

Content Filter User Interface

You can show the MailScan user interface by double clicking on the **Content Filtering** entry in the left-hand list of the WorkgroupMail administrator. When you do this the Settings dialog box is displayed.



More:

Rules

Adding a New Rule

Replacement fields

Rule ordering

Virus Checking using the Content Filter

Configuring the Profanity Checker

Rules

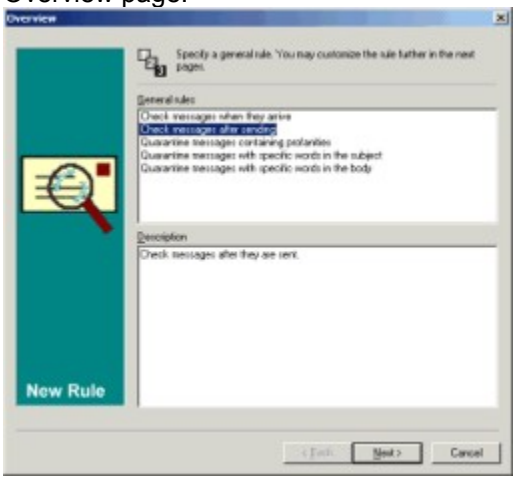
This dialog box shows the list of rules that are currently defined. The content filter is pre-configured with the following rules:

- Messages containing EXE attachments should be quarantined and the appropriate person informed.
- Messages containing more than 10 attachments should be quarantined and the appropriate person informed.
- Outgoing messages sent to external recipients that contain rude words should be quarantined and the appropriate person informed.

The list of rules that you enter into the content filter determines your e-mail policy. Each rule comprises an event (mail is sent and/or mail is received), a condition (e.g. message contains profanities) and an action (e.g. quarantine the message and inform someone).

Adding a New Rule

You can add a new rule by pressing the **Add…** button. The Rule wizard will be displayed, showing the Overview page.



Select the appropriate event from the **General Rules** list. You can choose from the following:

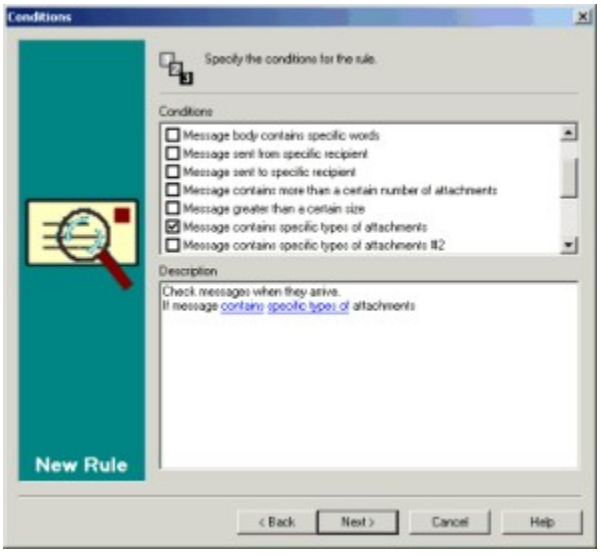
Events

Check messages when they arrive

Check messages before sending

Check all messages

The description of the event will appear in the Description window below. Press **Next** to specify conditions for the rule.



Select one or more conditions that must be satisfied in order for the rule to *fire*. You may choose from the following:

Conditions

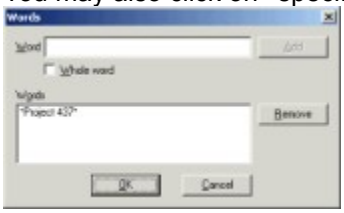
Message contains profanities

Message contains specific words

Message subject contains specific words

Message body contains specific words
Message sent from specific recipient
Message sent to specific recipient
Message contains more than a certain number of attachments
Message greater than a certain size
Message contains specific types of attachments
Message has attachments whose filename contains specific words
Message marked as specific priority
Message sent from MailScan
Within time range
MIME header field exists
MIME header field contains values(s)
Message is signed
Message sent to external recipient [Applies only to the event ‘Check messages before sending’]

You will notice that when you select certain conditions, the description shown in the window below will display blue underlined words. This represents data that you must specify. For example, if you select the condition "Message contains specific words ", then ‘‘contains’’ and "specific words" will appear underlined in the description window. You may click on ‘‘contains’’ and select either ‘‘contains’’ or ‘‘does not contain’’. You may also click on ‘‘specific words’’ and specify the exact words that you wish to match.

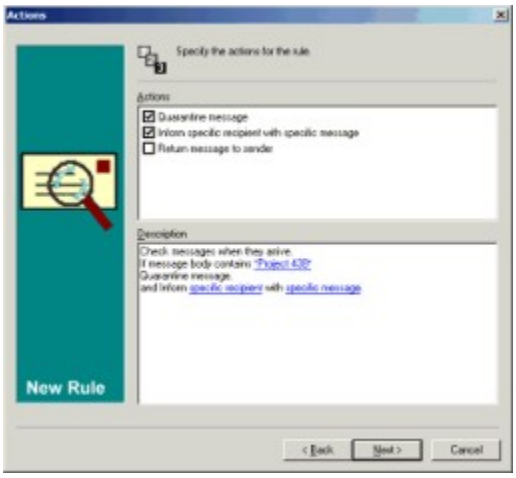


Enter the word or words that you want to match in the **Word** field and press the **Add** button. Press **OK** to save your changes. If you entered more than one word (i.e. more than one entry) then the condition will hold true if any of those words are contained in the message.

If you enter more than one condition for the rule then **all** conditions must be met in order for the rule to fire. If you wish to fire a rule based on one condition being met **or** another condition being met then you must achieve this by creating separate rules.

When you select certain conditions, another duplicate condition will appear in the conditions list. This lets you specify multiple conditions of the same type. For example, if you select the **MIME header field exists** condition you may want to also check for the existence of another field. The content filtering lets you do this by adding a **MIME header field exists #2** entry to the condition list, allowing you to specify another header that must exist in order for the rule to fire.

When you have specified the appropriate conditions, Press **Next** to specify what action should be taken when the conditions are met.



If the conditions of the rule are met, you may choose one or more of the following actions to be performed:

Actions

Quarantine the message

Inform specific recipient with specific message

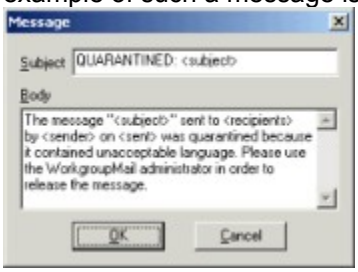
Return specific message to the sender

Inform intended recipients with specific message

Select the action(s) to be performed by ticking the appropriate tick boxes.

Replacement fields

When defining alerting messages in the Actions page, it is possible to use replacement fields in order to give the recipient of the alert some useful context. For example, if an administrator was to be alerted of an outgoing message being quarantined because it contained rude words, the administrator would need to know the subject of the outgoing message, who sent it, who it was to and possibly when it was sent. An example of such a message is shown below:



The following replacement fields may be used:

Replacement fields	Description
<Subject>	The subject of the message.
<Sender>	The sender of the message.
<SenderAddress>	The address of the sender of the message.
<Sent>	The date when the message was sent or received.
<Recipients>	The recipients of the message.
<RecipientCount>	The number of recipients to whom the message has been sent or received.
<Attachments>	A list of the attachment names in the message.
<AttachmentCount>	The number of attachments included in the message.

Rule ordering

The content filter lets you specify the order in which rules are processed. The ability to choose the order in which rules are processed in conjunction with the ‘‘Jump to rule’’ action gives you more control over how you deal with each sent or received message. For example, if you wanted to perform one action if a message contained a certain word in its subject and another action if the message did not contain the word, then you can define three rules as follows:

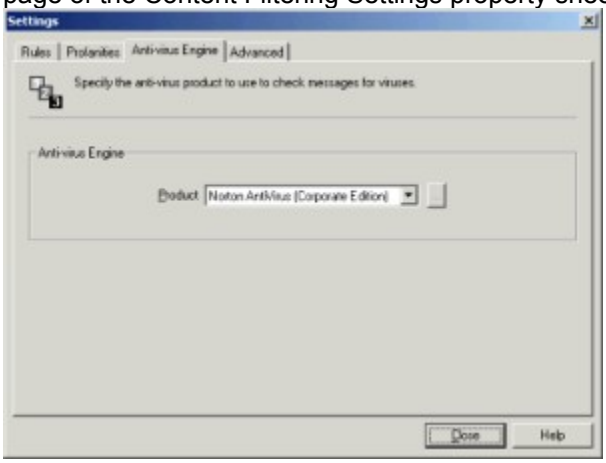
Rule	Event/Condition/Action
Rule 1	Check messages when they arrive. If message subject contains word then jump to rule 3.
Rule 2	Check messages when they arrive. Always do action A.
Rule 3	Check messages when they arrive. Always do action B.

Another use of rule ordering is where you wish to perform one action if conditions A and B are satisfied and another action if only condition A is satisfied. In this case, you simply order the rules so that the rule which checks for condition A and B comes above the rule which checks for condition A alone. To modify

the ordering of a rule, select the rule from the Rules page of the Content Filtering Settings property sheet and press the **Raise** or **Lower** buttons appropriately.

Virus Checking using the Content Filter

The content filter can check each incoming or outgoing message for viruses and perform any of its actions on encountering an infected message. To enable the content filter to check for viruses, you must select the appropriate anti-virus software that you wish the content filter to use in the Anti-virus Engine page of the Content Filtering Settings property sheet.

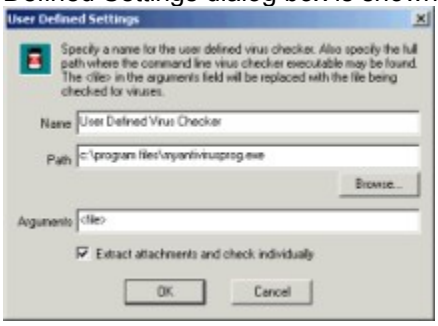


Then press the **Close** button at the bottom of the property sheet. When you next open the Content Filtering Settings property sheet and you add a new rule, you will notice that a new condition has been added called **message contains a virus**. You can now use this condition in conjunction with other conditions and rules in order to determine what will happen when a virus infected message is detected.

The Anti-virus Engine page lets you select from the following anti-virus products:

Program
Norton Antivirus (desktop edition)
Norton Antivirus (corporate edition)
McAfee VirusScan
Network Associates VirusScan
Sophos Antivirus

There is also an entry called **User defined…** in the drop down list. If your anti-virus software is not listed, then, provided your anti-virus software supports a command line interface, you can still use it in conjunction with WorkgroupMail by selecting the **User defined…** entry. When you do this the User Defined Settings dialog box is shown.

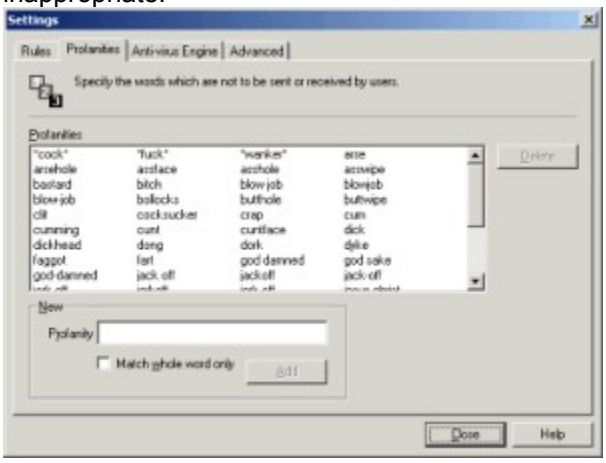


Enter the full path to your command line based anti-virus software program and by specifying any arguments that should be passed through to this program. By default, there is an argument called **<file>**. This is the file representation of the message that WorkgroupMail will pass to your anti-virus software and

must always appear in the **Arguments** field.

Configuring the Profanity Checker

The Profanities page in the Content Filter property sheet shows a list of unacceptable words that should not appear in a mail message. The content filter provides an action called **Message contains profanities**, which essentially means that the message contains one or more of the words listed in this page. You can change the words that appear in this list to reflect words that your organization deems inappropriate.

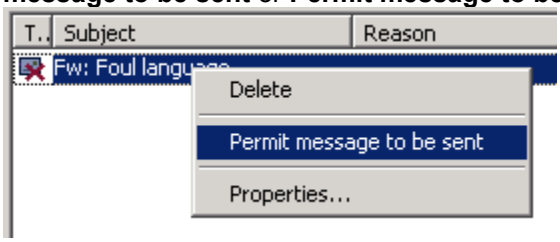


To add a new word, enter the word into the Profanitiy field and press the **Add** button. This will look for any occurrence of this string of characters anywhere inside the message. For example, if you entered the word ‘‘bum’’, any message containing the word ‘‘bumper’’ would be considered inappropriate. This may not always be desirable. If you wish to look for occurrences of the individual word ‘‘bum’’, you must tick the **Match whole word only** tick box before you press the **Add** button.

To delete a word, select it from the list and press the **Delete** button.

Quarantined Messages

Once your e-mail policy has been defined then if any messages are sent or received which violate the specified rules, the messages will be moved to the selected quarantine as appropriate and will not be sent or received. An administrator may view the messages in any quarantine and can permit the message to be released by clicking on the appropriate message using the right mouse button and selecting **Permit message to be sent** or **Permit message to be received**.



If the message should not be permitted, then the administrator can delete it by selecting **Delete** from the same context menu. For more information on quarantine areas, see Quarantine Areas.

Overview

The Virus Protection feature lets you virus check any incoming or outgoing message using your existing anti-virus software. The features of this component are available in the content filter, so this component is not needed if you already use the content filter.

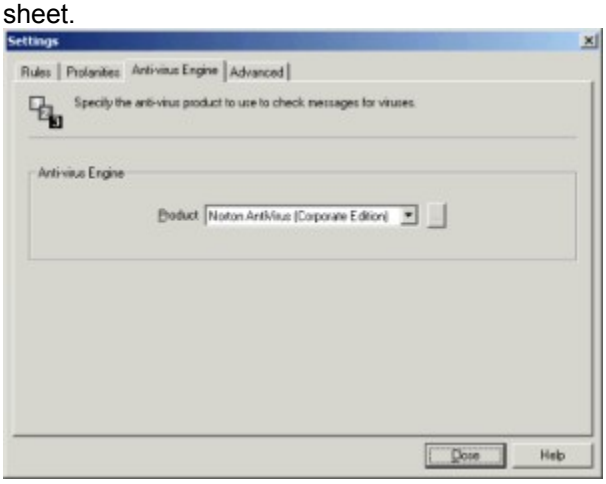
Virus Protection comes pre-installed in the Professional edition of WorkgroupMail and the Home edition of WorkgroupMail. It is not included in the Enterprise edition of WorkgroupMail because this edition includes the content filtering component, which includes the virus protection functionality.

Note: This component will function during the 30-day trial period, but must be purchased separately if used in the Professional edition of WorkgroupMail after the 30-day trial period.

Configuring Virus Protection

Double click on the Virus Protection entry. The Settings property sheet will be displayed.

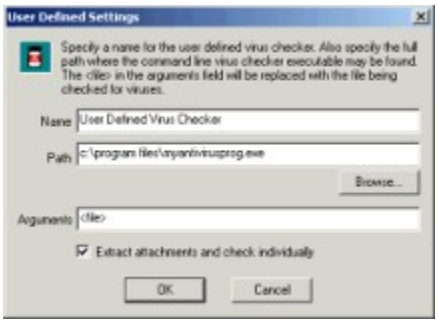
To enable the content filter to check for viruses, you must select the appropriate anti-virus software that you wish the content filter to use in the Anti-virus Engine page of the Content Filtering Settings property sheet.



The Anti-virus Engine page lets you select from the following anti-virus products:

Program
Norton Antivirus (desktop edition)
Norton Antivirus (corporate edition)
McAfee VirusScan
Network Associates VirusScan
Sophos Antivirus

There is also an entry called **User defined…** in the drop down list. If your anti-virus software is not listed, then, provided your anti-virus software supports a command line interface, you can still use it in conjunction with WorkgroupMail by selecting the **User defined…** entry. When you do this the User Defined Settings dialog box is shown.



Enter the full path to your command line based anti-virus software program and by specifying any arguments that should be passed through to this program. By default, there is an argument called **<file>**. This is the file representation of the message that WorkgroupMail will pass to your anti-virus software and must always appear in the **Arguments** field.

Once you have selected the relevant anti-virus software from the **Product** drop down list, select the quarantine to which infected messages should be sent. At the bottom of the page, tick the **Inform a user when a virus is detected** tick box and select a user who should be informed of viruses, if they are detected.

Press **OK** in the Settings property sheet. From now on, all incoming and outgoing messages will be checked using your antivirus software.

Overview

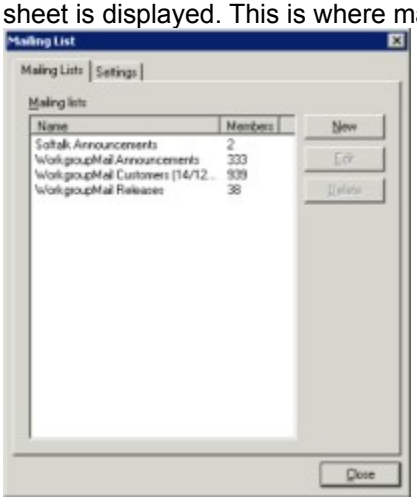
The Mailing List plug-in is a powerful tool that allows you to broadcast announcements to your customers and business contacts by sending one e-mail to a mailing list address. The membership of a mailing list can range from a handful to several thousand. Special offers, newsletters and forthcoming events can be sent to the members of your list with just one e-mail. You can also set up discussion lists where replies to the initial announcement are automatically circulated to all members.

You can publicise the mailing list on your website or in e-mail messages that you send. New members can join the list by completing an on-line form or by sending an e-mail to the mailing list with the word ‘‘subscribe’’ as the subject of the message. The Mailing List plug-in automates the subscription process by sending out a confirmation message, which the subscriber must respond to within a predefined time frame. This process is called *double opt-in.* This way your news is only circulated to those who have expressed a genuine interest. If a subscriber wishes to leave the mailing list, he must follow the instructions at the foot of the e-mail and the Mailing List plug-in will do the rest.

Configuring Mailing Lists

When the Mailing List plug-in is installed, the Mailing List Manager entry will be displayed in the left hand window of the Administrator. Selecting this entry will display the Mailing List Manager Summary page in the right hand window, which shows the registration status and the configured Mailing Lists.

The Mailing List plug-in can manage an unlimited number of mailing lists. To add a new mailing list, double click on the Mailing List icon in the left-hand list of the administrator. The Mailing List property sheet is displayed. This is where mailing lists are created, modified, deleted and administered.



Click the **New** button to create a new mailing list. Enter a name for the list and enter an e-mail address for the mailing list. This is the address that is used when sending a message to the mailing list and to which replies in a discussion are sent. This should be a valid address which is handled by WorkgroupMail, e.g. news@mydomain.com. Select a moderator from the list of WorkgroupMail users. This is the only person allowed to send an announcement or initiate a discussion as a non-member.

Select the type of mailing list that you wish to create. The WorkgroupMail Mailing List plug-in manages three types of mailing list. They are as follows:

- **Announcement List**

Only the moderator can send messages to an announcement list. A message sent to this type of list is sent to every member of the list. List members are not permitted to reply to any messages sent from this list.

- **Discussion List**

The moderator or any member of the list may send a message to this list. A message sent to this type of list is sent to every member of the list. List members may reply to messages sent from this list.

- **Open List**

Anyone may send messages to this list, including non-members. A message sent to this type of list is sent to every member of the list. List members may reply to messages sent from this list.

Press **OK** to create the mailing list. The mailing list is now live. Any messages sent to the list will now be sent to the members.

Adding members to the list

Members can automatically add themselves to a mailing list by sending a message to the list with 'subscribe' as the subject of the message. When they do this, the mailing list will send them a confirmation message, asking them to reply to the message in order to confirm that they are to become a member of the list.

If you wish to manually invite a person to a mailing list then select the Details tab of the appropriate Mailing List property sheet and press the **Invite** button. The Invite button, which is initially greyed out, is used to manually invite someone to join the mailing list. The button becomes active when the mailing list has been fully configured and saved.

If you wish to manually add a person to a mailing list, thereby skipping the confirmation process, then select the Members tab of the appropriate Mailing List property sheet and press the **New** button. The Name and Address fields will become enabled. Enter the person's name and e-mail address into these fields and press the **Add** button.

WorkgroupMail lets you import a list of users from a file. This file should be formatted as follows:

```
"John Smith"<john.smith@company.com>
"Paul Eacott"<paule@hotmail.com>
fred.jones@domain.com
sue.simmonds@yahoo.com
```

The name is optional, but if included, should appear in quotes. Each name/address pair should start on a new line.

If you wish to import a list of users then press the **Import** button. A file chooser dialog box is shown. Select the file containing the list of names and addresses and press **OK**.

Removing members from the list

Members of the list can automatically remove themselves from the list by sending a message to the mailing list with the word ‘unsubscribe’ as the subject of the message. When they do this they will receive a message confirming that they have been removed from the list.

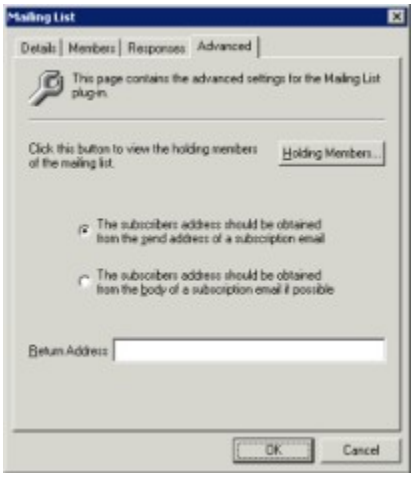
You can manually remove a member from a list by selecting the Members page of the appropriate mailing list property sheet, selecting the relevant member and pressing the **Delete** button.

Advanced functions

Holding members

When someone initially subscribes to a mailing list or when they are initially manually invited to the list, they are placed in the *holding members* list. They remain in this list until they confirm their membership or until they have been in the holding members list for longer than 10 days (the default holding duration). It is possible to modify the holding duration from the Settings page of the Mailing List property sheet.

You can view the people currently held in the holding members list by selecting the Advanced page of the appropriate mailing list property sheet and pressing the **Holding Members** button. A list of unconfirmed members is displayed. From here, you may delete any held members or convert them to full members of the list.



Determining a subscribers e-mail address

If someone attempts to join a mailing list by sending a message to the list with ‘subscribe’ as the subject line then the mailing list plug-in can determine the e-mail address of the subscriber by looking at the sender address of the message. However, if you are encouraging prospective members to join your list by entering their e-mail address into a form on a web page then, often, the required e-mail address will appear in the body of the message rather than as the sender of the message. For this reason, WorkgroupMail lets you choose how to determine the e-mail address of the subscribers to this particular list.

The return address

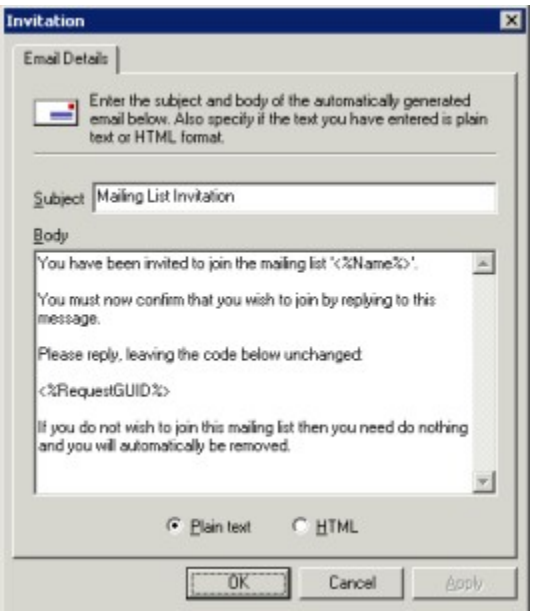
The Advanced page lets you specify a return address for a mailing list. This is used to specify the reply address when a message is broadcast to the members of a list. It does not affect automated replies to subscription messages. Specifying a return address that is different from the mailing list's e-mail address lets you avoid e-mail looping which can occur if the mailing list sends out a message which is automatically responded to by a mail server.

Customizing responses

For each mailing list, you can customize the responses that are sent to the subscriber or member. The following responses may be customized:

Response message	Purpose
Invitation message	This is the message that is sent to a prospective member when they are manually invited to the mailing list.
Request message	This is the message that is sent to a prospective member when they ask to join a list by sending a message to the list with 'subscribe' as the subject.
Welcome message	This is the message that is sent to a member when they successfully join the mailing list.
Unsubscribe message	This is the confirmation message that is sent to a member when they have successfully unsubscribed themselves from the mailing list.
Suffix text	This is the text that is appended to each broadcast message that gives instructions on how to unsubscribe from the list.

When you press one of the buttons, a dialog box is displayed, showing the subject and body of the message. At the bottom of the dialog box there are two radio buttons which let you choose whether the body of the message is HTML or plain text.



More:

Replacement fields

Replacement fields

Inside the body of each response message, you can include certain replacement fields in order to make each message more context aware. In the Mailing List Invitation and Request responses you must include the <%RequestGUID%> replacement field. WorkgroupMail will replace this field with a code which will enable it to identify the context of a response to one of its outgoing messages. The permitted replacement fields are as follows:

Replacement field	Description
<%RequestGUID%>	Replaced with a code which lets WorkgroupMail identify the originating member.
<%Name%>	Replaced with the name of the mailing list.
<%Address%>	Replaced with the address of the mailing list.

Configuring the Shared Address Book

The Shared Address Book component possible makes it possible for users to share and lookup names and email addresses from their mail client.

WorkgroupMail comes supplied with the shared address book component pre-installed. This feature is included in the Enterprise edition of WorkgroupMail. It is not available in the Home edition of WorkgroupMail.

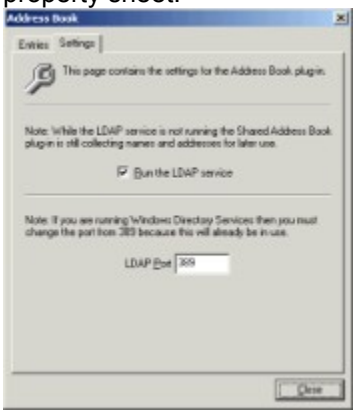
Note: This component will function during the 30-day trial period, but must be purchased separately if used in the Professional edition of WorkgroupMail after the 30-day trial period.

The Shared Address Book extracts the names and addresses from all incoming and outgoing messages and automatically adds the information to a server-based data store.

Once enabled, the Shared Address Book will expose the names and addresses via an LDAP server. This means that any LDAP client, such as Outlook Express or Outlook may use the Shared Address Book to lookup e-mail addresses from a name. Once the Shared Address Book is enabled, you will be able to open a new Compose Window in your mail client, and enter the first few letters of a person's name into the To: field and then press the **Check** button. The mail client will communicate with WorkgroupMail in order to determine the full name and e-mail address of the required recipient.

You can also add entries to the address book manually. To do this, press the **New** button. The Name and Address fields will be enabled. Enter a name and address into these fields and press the **Update** button. To delete an entry, select the entry in the list and press the **Delete** button.

By default, the Shared Address Book does not expose the names and addresses via an LDAP server. In order for your mail client to make use of the names and addresses in the Shared Address Book, you must first turn on the LDAP server. You can do this from within the Settings page of the Shared Address Book property sheet.



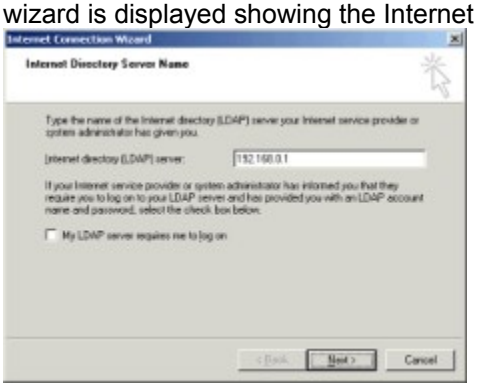
Tick the **Run the LDAP Service** tick box and press **Close**. Once you restart the WorkgroupMail server program, the names and addresses will be exposed via the LDAP server.

Note: Windows 2000 Server uses the LDAP protocol for the Windows directory. If you try to enable the Shared Address Book LDAP service on a Windows 2000 Server that is hosting the Windows directory, WorkgroupMail will refuse to start its LDAP server. Instead, you must change the LDAP port from 389 to 8389 and must make the corresponding change in the advanced properties of the LDAP settings on each mail client.

Configuring the Mail Clients

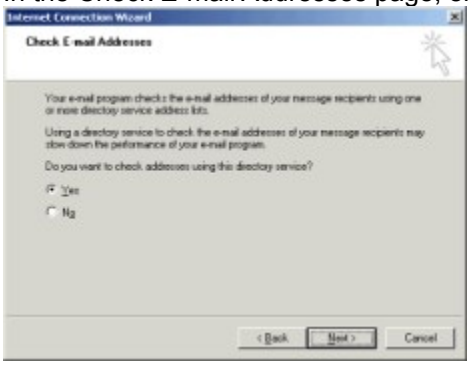
Configuring Outlook Express

To configure Outlook Express, select **Accounts** from the **Tools** menu and select the Directory Service page. Press the **Add** button and select **Directory Service…** from the sub menu. The Internet Connection wizard is displayed showing the Internet Directory Server Name page.



In the **Internet directory (LDAP) server** field, enter the IP address or name of the computer which runs WorkgroupMail. Ensure that the **My LDAP server requires me to log on** is NOT ticked. Press **Next** to continue.

In the Check E-mail Addresses page, ensure that you select the **Yes** radio button as shown below:



Press **Next** and then **Finish**. You will now be able to bring up a new Compose window and enter into the **To:** field the first few letters of the name of a recipient that exists in the Shared Address Book. When you press Ctrl+K, Outlook Express will lookup and complete the name, assigning it an e-mail address.

Configuring Outlook

Ensure that Outlook is configured for Corporate mode (for versions Outlook 2000 or less). Select **Services…** from the **Tools** menu. The Services dialog box will appear. Press the **Add…** button and select **Microsoft LDAP Directory** from the **Add Service to Profile** dialog box. Press **OK**. The LDAP Directory Service dialog box is displayed.

Enter the IP address of the computer which runs WorkgroupMail into the **Server Hostname** field. Leave all the other fields as shown above.

What Is IMAP?

IMAP stands for **I**nternet **M**essage **A**ccess **P**rotocol. It is a protocol that mail clients, such as Outlook or Outlook Express use for accessing messages held on a mail server. It differs from POP3 in that it enables a client to access and manipulate messages held on the server as if they were local. The idea behind POP3 is that the mail client transfers the messages from the server to its local message store. With IMAP, the messages and message folders always remain on server and are just accessed or modified from the mail client. The advantage of this is that it then becomes possible to access and manipulate mail from more than one computer or from more than one client.

When applied to WorkgroupMail, this means that it becomes possible to access the same messages and message folders from your mail client, such as Outlook or Outlook Express, and from WebMail.

When used in conjunction with the public folders functionality provided by WorkgroupMail, IMAP becomes a powerful methodology for sharing message folders.

Configuring IMAP

IMAP is available in both the Professional and the Enterprise editions of WorkgroupMail. Making use of IMAP is simply a case of configuring your mail client to have an IMAP account.

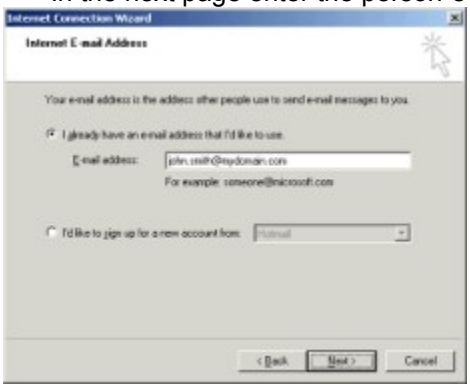
More:

Configuring Outlook Express

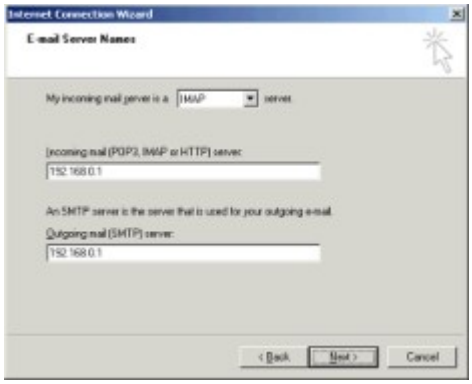
Configuring Outlook Express

To configure Outlook Express for IMAP, do the following:

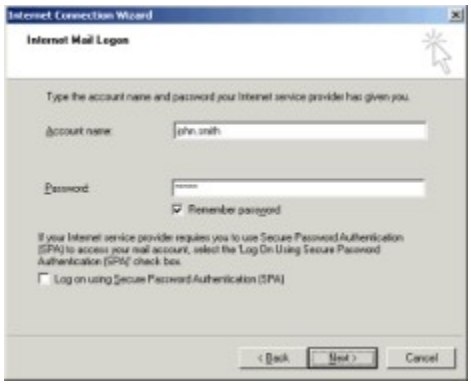
- In Outlook Express, select **Tools | Accounts…** and select the **Mail** tab.
- Delete any unused mail accounts that exist by selecting them and pressing the **Remove** button.
- Press the **Add -> Mail…** button. The Internet Connection wizard will appear.
- In the Display name field, enter the full name of the person whose mail client you are configuring and press **Next**.
- In the next page enter the person's e-mail address into the **E-mail address** field.



- Press **Next** to continue. In the **E-mail Server Names** page, specify that the incoming mail server is an **IMAP** mail server and enter the IP address of the computer which runs WorkgroupMail into the **Incoming Mail (POP3, IMAP or HTTP) Server** and **Outgoing Mail (SMTP) Server** fields.



- Press **Next** to continue. In the **Internet Mail Logon** page, enter the Local Account Name and Local Account Password, for the person whose mail client you are configuring, into the **Account name** and **Password** fields, respectively.



You can find out the Local Account Name and Local Account Password by selecting the appropriate user in the left-hand list of the WorkgroupMail administrator and viewing the appropriate information in the right-hand window. Press **Next** to continue. This will take you to the **Finish** page. Click on the **Finish** button to complete the configuration. When you do this Outlook Express will ask you if you want to download the folders for this account. Press **Yes**. Provided that the WorkgroupMail server program is running, Outlook Express will download the person's folders making all their email available to the mail client.

Overview

WebMail for WorkgroupMail provides you with the ability to access your inbox and folders using Internet Explorer. It is an ASP application that must be installed onto Windows 2000/XP running IIS and WorkgroupMail. WebMail is a free add-in that will work with the Professional edition and the Enterprise edition of WorkgroupMail. It will not work with the Home edition of WorkgroupMail.

Installing WebMail

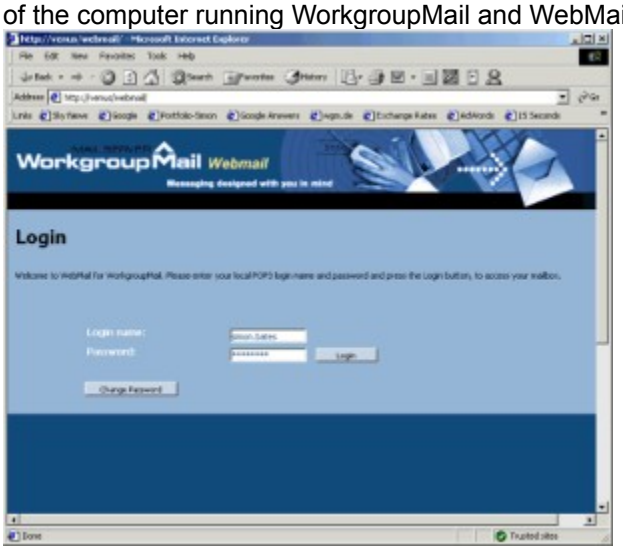
Once you have downloaded the **webmail.exe** self extracting executable file, run it by opening or double clicking on the file. The WinZip self extracting dialog box is displayed.

Press **Setup** to start the installation program, showing the Welcome page. Press **Next** to show the license page, which you must agree to by pressing the **Accept** button. When you press the **Accept** button, WebMail setup will copy the relevant files to the **program files\workgroupmail\data\webmail** folder and will create a **wgmweb** virtual directory and will modify the access rights of the IUSR_computername and IWAM_computername users to have full access to the WorkgroupMail data folder.

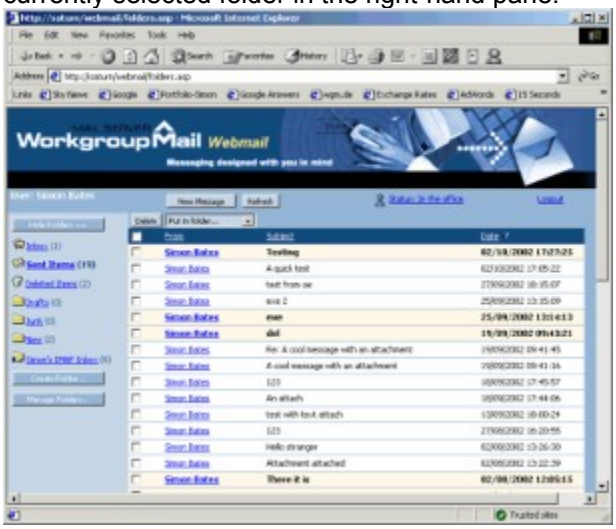
Logging in to WebMail

In order for a user to log into WebMail, they must know their local account name and password. This information may be found by selecting the appropriate user from the left-hand window of the WorkgroupMail Administrator and viewing the **Local account name** and **Local password** settings in the right-hand window. If the user was imported from the active directory, the local account name and local password for that user will be that user's Windows login name and password.

To login to WebMail, browse to **http://computername/wgmweb**, replacing *computername* with the name of the computer running WorkgroupMail and WebMail. The login page is shown.



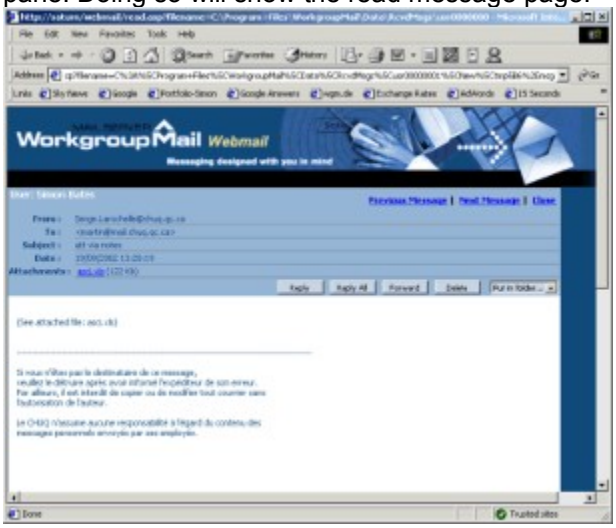
Enter the appropriate login name and password and press the **Login** button. The folders page is shown, displaying the users local folders and all public folders in the left hand pane and any messages in the currently selected folder in the right-hand pane.



You can select a folder by clicking on it in the left-hand pane. When you do so, a list of the messages in that folder is displayed in the right-hand pane.

Opening a Message

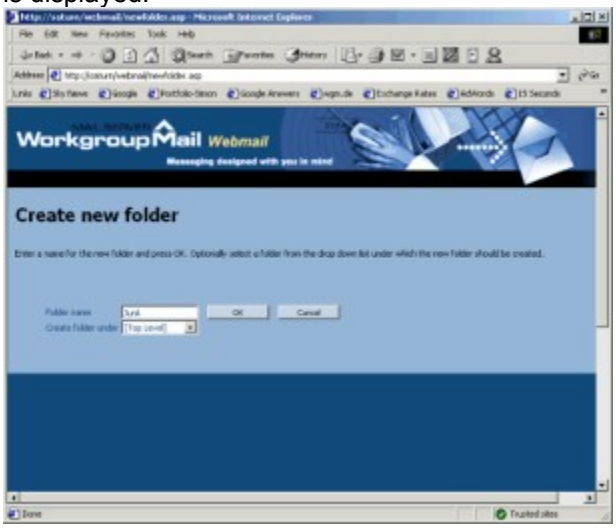
You can view a message by clicking on the blue underlined sender of the message in the right-hand pane. Doing so will show the read message page.



You can reply to the message or forward the message by pressing the **Reply**, **Reply All** or **Forward** buttons in the header of the message. You can delete the message by pressing the **Delete** button, or you can file the message to a different folder by selecting the appropriate folder from the drop down list. To close the message, press the **Close** link at the top right of the page.

Creating a New Personal Folder

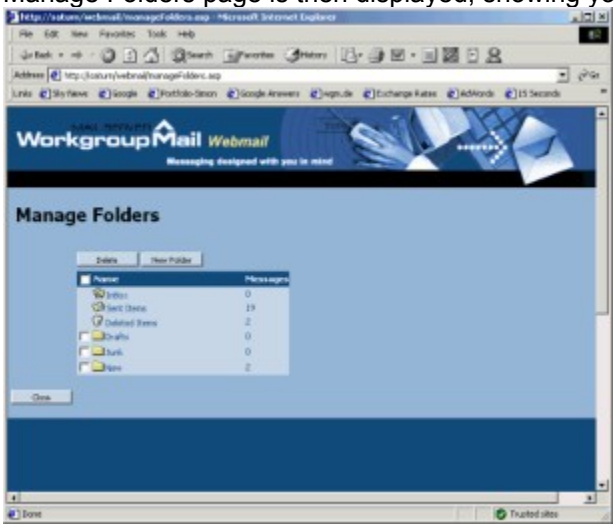
To create a new personal folder, press the **Create Folder** button in the folders page. A New Folder page is displayed.



Enter a name for the new folder in the **Folder Name** field. If you want to create this folder as a sub folder of an existing folder, select the appropriate parent folder in the **Create folder under** field. If you want the new folder to be a top level folder then ensure that [Top Level] is selected. Press **OK** to create the new folder.

Deleting Personal Folders

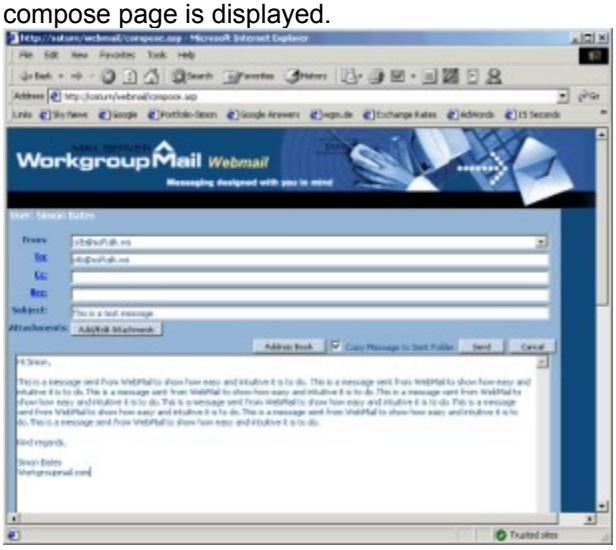
You can delete existing personal folders by pressing the **Manage Folders** button in the folders page. The Manage Folders page is then displayed, showing you a list of existing public folders.



Select any folders that you wish to delete by ticking the adjacent tick box. Then press the **Delete** button, situated above the list of folders. This page also lets you create a new folder in the same way as described in the previous section.

Composing a New Message

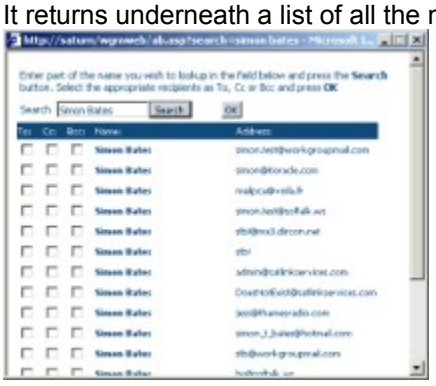
WebMail lets you send messages in just the same way that you would do from any mail client. To compose a new message, press the **New Message** button, located at the top of the folders page. The compose page is displayed.



Enter the e-mail address of any recipients that you wish to send to in the To:, Cc: or Bcc: fields as appropriate. If you want to use the address book to look up email addresses by name then press the **Address Book** button to show the address book.

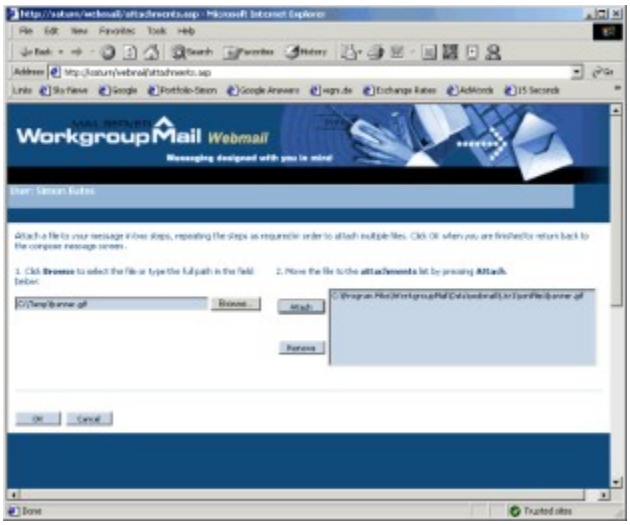
Note: In order to use the address book, you must be licensed to use the Shared Address Book plug-in.

The address book lets you enter any part of the recipient's name or e-mail address into the **Search** field. It returns underneath a list of all the matches that it found.



You can select any of the results to appear in the To:, Cc:, or Bcc: field of the Compose window by selecting the appropriate tick box to the left of the relevant result.

Once you have specified the recipients of the message, enter a subject for the message and some body text. If you want to add one or more attachments to the message, press the **Add/Edit Attachments** button. The attachments page is displayed.

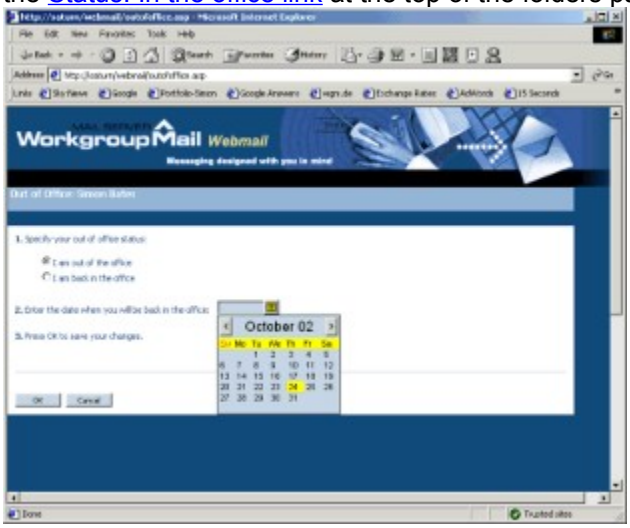


Use the **Browse** button to locate the file that you wish to attach to the message. Then press the **Attach** button to add this file to the list of attachments for the message. Repeat this until you have attached all the necessary attachments. Then press **OK**. The file attachments will appear to the right of the **Add/Edit Attachments** button. You can add or remove attachments from the message by pressing the **Add/Edit Attachments** at any time prior to sending the message.

When you are ready to send the message, press the **Send** button located to the right of the compose window. If you wish to keep a copy of the message in the Sent Items folder, ensure that you tick the **Copy message to sent folder** tick box.

Out of Office

WorkgroupMail lets any WebMail user specify whether or not they are away from the office. When a user is marked as away from the office, WorkgroupMail will automatically respond to any messages sent to that user, notifying the sender that the user is not in the office and informing them when the user will be returning to the office. In order for a user to specify that they are away from the office, they should click the Status: In the office link at the top of the folders page. The Out of Office page is displayed.



When the user selects the **I am out of the office** radio button, they will be prompted to specify the date that they are due back in the office. The date that they specify must be a future date. When they press **OK**, WorkgroupMail will have marked them as away from the office. WorkgroupMail will automatically mark the user as back in the office on the date that they are due back. The user can mark themselves as being back in the office at any time by clicking on the same status link and selecting the **I am back in the office** radio button in the Out of Office page and pressing **OK**.

Overview

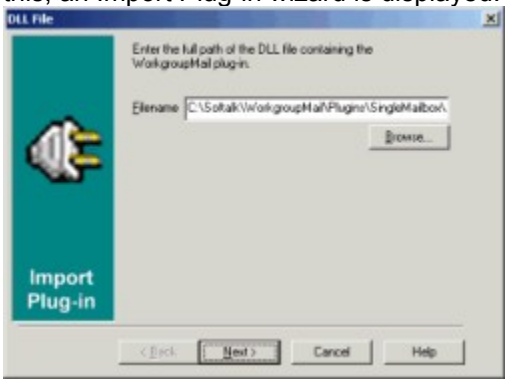
WorkgroupMail is an extensible product in that software plug-ins may be imported into WorkgroupMail in order to perform specific processing of incoming and outgoing messages. This makes it possible to incorporate powerful message screening or processing into the standard WorkgroupMail user interface.

To view the available plug-ins, visit http://www.workgroupmail.com .

You can develop your own plug-ins to perform your own specific message processing prior to delivery of incoming messages and prior to sending of outgoing messages. Plug-ins can be developed in Visual Basic or Visual C++. For more information, visit http://www.workgroupmail.com .

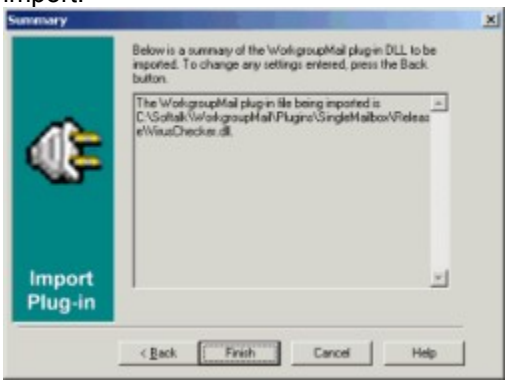
Installing a Plug-in

WorkgroupMail plug-ins come in the form of a DLL file. You can import a plug-in into WorkgroupMail by selecting **Import Plug-in** from the **File** menu of the WorkgroupMail administrator program. When you do this, an Import Plug-in wizard is displayed.



Use the **Browse…** button to locate the plug-in DLL file or enter it directly into the Filename field and press **Next** to continue.

The last page shown is the Summary page which confirms the filename of the plug-in you are about to import.



Press **Finish** to import the plug-in.

The plug-in will appear in the WorkgroupMail administrator as an entry in the left-hand window above the Event Log.

More:

Editing a Plug-ins Properties

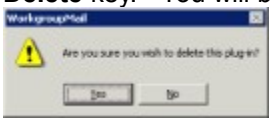
Deleting a Plug-in

Editing a Plug-ins Properties

You can edit the properties of the plug-in by double clicking on the plug-in using the left mouse button. The user interface that appears when you do this is specific to the plug-in. See the relevant plug-in help for more information.

Deleting a Plug-in

You can delete a plug-in by selecting it in the WorkgroupMail administrator program and pressing the **Delete** key. You will be asked to confirm that you wish to delete the plug-in.



Press **Yes** to delete the plug-in.

Overview

Developers can extend and tailor WorkgroupMail to specific requirements in two ways:

- Plug-in development
- WorkgroupMail API

By developing plug-ins, it is possible to change the way WorkgroupMail processes inbound and outbound messages. By using the WorkgroupMail API, it is possible to make programmatic changes to the administrative settings in WorkgroupMail and to programmatically send messages through WorkgroupMail.

Developing Plug-Ins

WorkgroupMail has the ability to add software plug-ins which can perform processing on incoming and outgoing messages before they are sent or received. This powerful feature makes it possible for you to tailor WorkgroupMail for a particular need. Plug-ins may be written in Visual C++ or in Visual Basic or in any language that supports COM and/or automation. Visual Basic is the preferred method and this section describes the process of writing such an application using Visual Basic as an example.

Whenever a message is sent or received, a certain method (or function) in your program will be called, passing you the name of a file which contains the message in its entirety in MIME format. You may do anything within this function, including deleting this file, moving it to another user, or mailing it externally. What you return from this function determines whether or not WorkgroupMail sends the message to the message quarantine.

When your plug-in is imported into the WorkgroupMail administrator, it will be visible as an entry in the left-hand list. When a user double clicks on the plug-in entry, another function will be called in your program, enabling you to show whatever user interface you wish. This may be anything from a message box, to a multi-page property sheet to a full blown application.

More:

Getting Started

The Programmatic Interface

Importing the Plug-in

Getting Started

A good starting point is to use the sample Visual Basic plug-in that you can download from **http://www.workgroupmail.com/sample_plugin.zip** . This is a Visual Basic project containing all the necessary functions to work with the WorkgroupMail engine and the WorkgroupMail Administrator.

Once you have downloaded the sample, load it into Visual Basic. You will see four functions defined. These are described in the next section:

The Programmatic Interface

There are four functions that provide the link between WorkgroupMail, the WorkgroupMail administrator and your plug-in program. These are as follows:

EditProperties()

```
Syntax: EditProperties( )
```

This function is called whenever the plug-in entry is double clicked in the WorkgroupMail administrator. This function takes no arguments and returns nothing back to the caller.

Ideally, this function should create a modal dialog box or property sheet, allowing the user to change any settings associated with the plug-in.

DefaultName()

```
Syntax: DefaultName( ) As String
```

This function is called by the WorkgroupMail Administrator in order to query the name of the plug-in. The plug-in name will be used to display the plug-in in the left hand window in the WorkgroupMail Administrator.

So, for example, you might return a typical DefaultName string as follows:

DefaultName = "Sample Plug-in"

GetProperties()

```
Syntax: GetProperties() As String
```

This function is used to display the properties (or settings) associated with the plug-in, in the right hand window of the WorkgroupMail Administrator when the user clicks on the plug-in entry in the left-hand window.

The string that you return from this function represents all the properties and values of the plug-in. The format of the string is as follows:

Property 1 & Value 1 & Property 2 & Value 2….

So, for example, you might return a typical property string as follows:

GetProperties = "Name&Sample Plug-in&Folder&" & "c:\temp" & "&Date&" & "12/4/01"

ProcessMessageFile()

```
Syntax: ProcessMessageFile(sFilename As String,
        bReceived As Integer, ByRef sDescription As
        String ) As Long
```

This function is called by the WorkgroupMail engine to determine whether or not a particular message should be quarantined.

The ProcessMessageFile() should do the following:

- Parse the file, sFilename, and look at whether the message is incoming or outgoing (bReceived) to determine whether this plug-in should quarantine or not.
- If the message is quarantined remember to assign a reason string to the sDescription parameter.
- Return one of the following values to determine how the message is treated by the engine:

Return Value	Action
1	Do not quarantine the message. Don't pass this message to

	any further plug-ins and don't permit the message to be sent or received.
2	Quarantine the message. Don't pass this message to any further plug-ins and don't permit the message to be sent or received.
3	Do not quarantine the message. Pass this message to any further plug-ins for processing. Permit this message to be sent and received unless another plug-in throws a problem.
4	Log an event to indicate a problem with the plug-in. Pass this message to any further plug-ins for processing. Permit this message to be sent and received unless another plug-in throws a problem.

Importing the Plug-in

To import the plug-in, select **Import Plug-in** from the **File** menu of the WorkgroupMail administrator. In the first page of the Import Plug-in Wizard, enter the location of the plug-in DLL. In the next page, enter the progid of the plug-in, which in the case of the sample plug-in is **sampleplugin.engine** . Press **Finish** to complete the import.

Overview

The WorkgroupMail API provides programmatic control over some of the features in WorkgroupMail. Most of the objects and methods relate to the administration of WorkgroupMail, however there are also specific methods for creating and sending messages programmatically via WorkgroupMail. The WorkgroupMail API is made up of the following automation objects.

Object Name	Description
WMSession	The object, which represents a login session with WorkgroupMail.
WMDomain	Represents a domain.
WMGroup	Represents a group of users.
WMISP	Represents an ISP
WMMailbox	Contains methods and properties that are common between the WMUser and the WMVirtualMailbox objects.
WMPublicFolder	Represents a public folder.
WMQuarantineArea	Represents a quarantine area.
WMUser	Represents a WorkgroupMail user.
WMVirtualMailbox	Represents a virtual mailbox.
WMUserDomainSettings	Represents a link between the user and the domain.
WMUserDomainSettingsList	A list of WMUserDomainSettings objects.
WMUserISPSettings	Represents a link between the user and an ISP.
WMAddressOrRange	Represents an IP address or a range of IP addresses.
WMAttachment	Represents a message attachment.
WMComposeMessage	Represents a new message to be sent.
WMFolder	Represents a personal folder.
WMMessage	Represents an e-mail message.
WMMessageList	A list of WMMessage objects.
WMProperty	Represents a user defined property that may be stored with the WMISP, WMUser and WMVirtualMailbox objects.
WMSchedule	An object which represents the times of day and days of the week when WorkgroupMail should connect to an ISP in order to send and receive mail.
WMSpamServer	Represents an entry for a anti-spam server, such as ORDB or MAPS.

WMSession

WMSession is the object that represents a login session with WorkgroupMail. The caller must log on to a session, either as an administrator, or as a user, before being permitted to perform most other operations. The WMSession object is the only object that needs to be explicitly created. All other objects may be accessed from this object. The following code is an example of how to create a WMSession object in Visual Basic:

```
Set session = CreateObject("WMAPI.WMSession")
bSuccess = session.Login(“adminpassword”)
```

More:

Properties

Methods

Properties

LocalPOP3LoginName

Description: The local POP3 login name for the workgroup. This is the login name for an account which exposes all the messages held in WorkgroupMail for download.

Type: String

LocalPOP3Password

Description: The local POP3 password for the workgroup. This is the password for an account which exposes all the messages held in WorkgroupMail for download.

Type: String

UnknownRedirectUser

Description The WMUser object that unknown mail is forwarded to.

Type: WMUser object

WorkgroupName

Description: The Company Name as entered with the keycode. This is used only to verify the keycode.

Type: String

Administrator

Description: The WMUser object that is the administrator of the workgroup. This is the first user that was created when WorkgroupMail was installed.

Type: WMUser object

LogCommunications

Description: Specifies whether or not all SMTP, POP3 and IMAP communications should be logged to a file.

Type: Boolean

LogFile

Description: The full path and name of the file used to log all SMTP, POP3 and IMAP communications.

Type: String

ClientConnectionInterface

Description: The interface that the POP3 server listens on, for client connections. It takes the form of an IP address in dot notation . It must match one of the local IP addresses of the computer, which runs WorkgroupMail.

Type: String

ServerConnectionInterface

Description: The interface that the SMTP server listens on, for client connections (or other mail servers). It takes the form of an IP address in dot notation. It must match one of the local IP addresses of the computer, which runs WorkgroupMail.

Type: String

IMAPConnectionInterface

Description: The interface that the IMAP server listens on, for client connections (or other mail servers). It takes the form of an IP address in dot notation. It must match one of the local IP addresses of the computer, which runs WorkgroupMail.

Type: String

PurgeEventLogPeriodically

Description: Specifies if WorkgroupMail should periodically purge entries from the event log. This represents the corresponding UI settings in the Purging page of the Settings property sheet.

Type: Boolean

PurgeEventsAge

Description: The age, in days, that event log entries need to be before they are purged. This represents the corresponding UI settings in the Purging page of the Settings property sheet.

Type: Long Integer

DontDownloadLargeMessages

Description: Specifies if large messages are to be left on the server when downloading mail from ISPs. Represents the corresponding UI settings in the Advanced page of the Settings property sheet.

Type: Boolean

LargeMessageSize

Description: The size, in bytes, of the largest message that may be downloaded. Represents the corresponding UI settings in the Advanced page of the Settings property sheet.

Type: Long Integer

OpenRelay

Description: Sets or gets the open relay property of WorkgroupMail. Returns True if WorkgroupMail is configured as an open

relay. Allows the caller to configure WorkgroupMail as an open relay.

Type: Boolean

RelayAuthenticated

Description: Specifies if mail can be relayed using SMTP authentication. This option is ignored if OpenRelay is true. If this property is true then, if the local sender has authenticated themselves, using SMTP authentication, they will automatically be permitted to relay mail to addresses outside the domain.

Type: Boolean

AQLimitStoredMessages

Description: A boolean that specifies if a default limit will be imposed on the maximum number of messages allowed in a user account mailbox. User account quotas will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Quotas page of the Settings property sheet.

Type: Boolean

AQStoredMessageLimit

Description: The default account quota for the maximum number of messages that can be stored in a user account mailbox. Dependant on AQLimtStoredMessages being true. User account quotas will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Quotas page of the Settings property sheet.

Type: Long Integer

AQLimitStorageSize

Description: A global boolean setting that specifies if a limit will be imposed on the maximum amount of disk space that can be occupied by messages in a user account mailbox. User account quotas will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Quotas page of the Settings property sheet.

Type: Boolean

AQStorageSizeLimit

Description: The default account quota, in Kb, for the maximum amount of disk space that can be occupied by messages in a user account mailbox. Dependant on

AQLimtStorageSize being true. User account quotas will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Quotas page of the Settings property sheet.

Type: Long Integer

APAutoDeleteInactiveAccount

Description: A global boolean setting that specifies if inactive accounts will be deleted automatically. Specific user account pruning will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the Settings property sheet.

Type: Boolean

APInactiveDays

Description: The default number of days that an account has to be inactive before it is automatically deleted . Dependent on APAutoDeleteInactiveAccount being true. Specific user account pruning will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the Settings property sheet.

Type: Long Integer

APDeleteInboxMessages

Description: A global boolean setting that specifies if old messages in the inbox will be purged. The Inbox is where WorkgroupMail initially delivers new messages for a particular user. Specific user account pruning will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the Settings property sheet.

Type: Boolean

APDeleteInboxMessageDays

Description: The default number of days old that a message must be before it is automatically purged from the Inbox. Dependent on APDeleteInboxMessages being true. Specific user account pruning will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the Settings property sheet.

Type: Long Integer

APDeletePersonalFolderMessages

Description: A global boolean setting that specifies if old messages in any personal folders will be purged. Personal folders are folders that are created in WebMail or using an IMAP client. Specific user account pruning will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the Settings property sheet.

Type: Boolean

APDeletePFMessageDays

Description: The default number of days old that a message must be before it is automatically purged from any personal folder. Dependent on APDeleteInboxMessages being true. Specific user account pruning will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the Settings property sheet.

Type: Long Integer

APDeleteDeletedMessages

Description: A global boolean setting that specifies if deleted messages will be purged. Deleted messages are messages that exist in the Deleted Items folder. Specific user account pruning will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the Settings property sheet.

Type: Boolean

APDeleteDeletedMessageDays

Description: The default number of days old that a deleted message must be before it is automatically purged from an account. Dependent on APDeleteDeletedMessages being true. Specific user account pruning will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the Settings property sheet.

Type: Long Integer

ARCanUseWebMail

Description: A global boolean setting that specifies whether or not a user can use WebMail. Specific user account restrictions will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Restrictions page of the

Settings property sheet.

Type: Boolean

ARCanUseIMAP

Description: A global boolean setting that specifies whether or not a user can use IMAP. Specific user account restrictions will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Restrictions page of the Settings property sheet.

Type: Boolean

ARCanSendExternalMessages

Description: A global boolean setting that specifies whether or not a user can send external mail. If false, the exceptions list will prevail. Specific user account restrictions will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Restrictions page of the Settings property sheet.

Type: Boolean

ARCanReceiveExternalMessages

Description: A global boolean setting that specifies whether or not a user can receive external mail. If false, the exceptions list will prevail. Specific user account restrictions will override this setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Restrictions page of the Settings property sheet.

Type: Boolean

SpamFilterType

Description: Specifies the action to perform on detection of a spam message. This corresponds to the user interface in the Settings page of the Spam Filtering property sheet. The following values are valid.

0. Do nothing
1. Filter spam and place in specified quarantine
2. Filter spam and delete/reject messages.
3. Add a header to the message for subsequent detection in the content filter (if appropriate).

Type: Short integer

SpamFilterDefaultMessageType

Description: Sets or gets how WorkgroupMail should treat messages in terms of spam mail. The following values are valid.

0. No junk mail filtering

1. Treat all mail as junk mail except for white list.

2. Treat all mail as non-junk unless specifically identified as such by the black list, the spam servers or SpamCleanser.

Type: Short integer

SpamQuarantine

Description: Sets or gets the quarantine area that is used to hold messages that are detected as spam messages.

Type: WMQuarantineArea object

SMTPServerPort

Description: Sets or gets the port used for listening for incoming SMTP connections. This corresponds to the user interface in the Advanced page of the Settings property sheet.

Type: Short integer

POP3ServerPort

Description: Sets or gets the port used for listening for incoming POP3 connections. This corresponds to the user interface in the Advanced page of the Settings property sheet.

Type: Short integer

IMAPServerPort

Description: Sets or gets the port used for listening for incoming IMAP connections. This corresponds to the user interface in the Advanced page of the Settings property sheet.

Type: Short integer

DefaultOutOfOfficeReply

Description: Sets or gets the global out of office reply. This setting can be overridden on a per-user basis.

Type: String

Methods

Login

Description: Logs into the WorkgroupMail session. Most operations cannot be performed unless this function has successfully been called. The administrator password may be changed once successfully logged in.

Syntax: Login (sPassword as string)

Parameters: sPassword is the administrator password.

Return type: Boolean

Example:

```
bLoggedIn = session.Login("AdminPwd")
```

Logout

Description: Log out of the WorkgroupMail session.

Syntax: Logout()

Parameters: None

Return type: None

SetAdministratorPassword

Description: Set the administrator password. You must be logged in for this method to work.

Syntax: SetAdministratorPassword (sNewPassword as string)

Parameters: sNewPassword is the new password string.

Return type: Boolean indicating if the call was successful. If not logged into the session then false will be returned.

Reload

Description: Reloads the WorkgroupMail data which will update the data stored in the API's memory. This function can be used when a change to the data has been made by another instance of the API or the WorkgroupMail administrator.

Syntax: Reload()

Parameters: None

Return type: Boolean indicating success or failure.

AddUser

Description: Adds a user with the given name to the WorkgroupMail list of users. You must be logged in for this method to work.

Syntax: AddUser(sName as string)

Parameters: sName is the string that specifies the name of the user to be added.

Return type: WMUser object that is the user added. Returns nothing if an error occurs such as not logged into the session.

Example:

```
set user = session.AddUser("Paul Kennedy")
```

GetFirstUser

Description: Gets the first user in the list of WorkgroupMail users. You must be logged in for this method to work.

Syntax: GetFirstUser()

Parameters: None

Return type: WMUser object of the first user. Returns nothing if not logged into the session.

GetNextUser

Description: Get the next user in the list of WorkgroupMail users after the user passed into this method. Returns nothing if the user passed in is nothing or the last user You must be logged in for this method to work.

Syntax: GetNextUser(user as object)

Parameters: user is a WMUser object returned by a previous call to GetFirstUser() or GetNextUser().

Return type: WMUser object of the next user. Returns nothing if there is no next user.

DeleteUser

Description: Delete the given user from the list of WorkgroupMail users. You must be logged in for this method to work.

Syntax: DeleteUser(user as object)

Parameters: user is a WMUser object returned by a previous method call.

Return type: Boolean indicating if the delete was successful or not.

AddISP

Description: Add a new ISP to WorkgroupMail with the given name You must be logged in for this method to work.

Syntax: AddISP(sName as string)

Parameters: sName is a string specifying the name of the new ISP.

Return type: WMISP object which is the newly added ISP.

GetFirstISP

Description: Gets the first ISP in the list of WorkgroupMail ISPs. You

must be logged in for this method to work.

Syntax: GetFirstISP()

Parameters: None

Return type: WMISP object of the first ISP. Returns nothing if not logged into the session.

GetNextISP

Description: Get the next ISP in the list of WorkgroupMail ISPs after the ISP passed into this method. Returns nothing if the ISP passed in is nothing or the last ISP. You must be logged in for this method to work.

Syntax: GetNextISP (isp as object)

Parameters: isp is a WMISP object returned by a previous call to GetFirstISP() or GetNextISP().

Return type: WMISP object of the next ISP. Returns nothing if there is no next ISP.

DeleteISP

Description: Delete the given ISP from the list of WorkgroupMail ISPs You must be logged in for this method to work..

Syntax: DeleteISP(isp as object)

Parameters: isp is a WMISP object returned by a previous method call.

Return type: Boolean indicating success or failure.

AddVirtualMailbox

Description: Add a new virtual mailbox to WorkgroupMail with the given name. You must be logged in for this method to work.

Syntax: AddVirtualMailbox(sName as string)

Parameters: sName is the string specifying the name of the new virtual mailbox.

Return type: WMVirtualMailbox object which is the newly added virtual mailbox.

GetFirstVirtualMailbox

Description: Gets the first virtual mailbox in the list of WorkgroupMail virtual mailboxes. You must be logged in for this method to work.

Syntax: GetFirstVirtualMailbox()

Parameters: None

Return type: WMVirtualMailbox object of the first virtual mailbox. Returns nothing if not logged into the session.

GetNextVirtualMailbox

Description: Get the next virtual mailbox in the list of WorkgroupMail virtual mailboxes after the virtual mailbox passed into this method. Returns nothing if the virtual mailbox passed in is nothing or the last virtual mailbox. You must be logged in for this method to work.

Syntax: GetNextVirtualMailbox(vmb as object)

Parameters: vmb is a WMVirtualMailbox object returned by a previous call to GetFirstVirtualMailbox() or GetNextVirtualMailbox().

Return type: WMVirtualMailbox object of the next virtual mailbox. Returns nothing if there is no next virtual mailbox.

DeleteVirtualMailbox

Description: Delete the given virtual mailbox from the list of WorkgroupMail virtual mailboxes. You must be logged in for this method to work.

Syntax: DeleteVirtualMailbox(vmb as object)

Parameters: vmb is the WMVirtualMailbox object to be deleted.

Return type: Boolean indicating success or failure.

AddJunkMailEntry

Description: Adds a mail address to the black list of junk mail addresses. You must be logged in for this method to work.

Syntax: AddJunkMailEntry(sJunkMailAddress as string)

Parameters: sJunkMailAddress is a string which is a junk mail address.

Return type: Boolean indicating success or failure.

GetFirstJunkMailEntry

Description: Gets the first junk mail address in the black list. You must be logged in for this method to work.

Syntax: GetFirstJunkMailEntry()

Parameters: None

Return type: String which is the first junk mail address.

GetNextJunkMailEntry

Description: Gets the next junk mail address after the junk mail address passed in. You must be logged in for this method to work.

Syntax: GetNextJunkMailEntry(sJunkMailAddress as string)

Parameters: sJunkMailAddress is a junk mail address string which is

got from a previous call to GetFirstJunkMailEntry() or GetNextJunkMailEntry().

Return type: String which is the next junk mail address.

DeleteJunkMailEntry

Description: Deletes the junk mail address passed in. You must be logged in for this method to work.

Syntax: DeleteJunkMailEntry(sJunkMailAddress as string)

Parameters: sJunkMailAddress is the junk mail address string to be deleted.

Return type: Boolean indicating success or failure.

DisableSave

Description: Disables saving changes to the WorkgroupMail data store. To enable changes to be saved you must call EnableSave() some time after this method call. This function might be called if a lot of changes are to be made so they can be made quickly without updating the data store and then the data store is updated when EnableSave() is called.

Syntax: DisableSave()

Parameters: None

Return type: None

EnableSave

Description: Enables saving changes to the WorkgroupMail data store. If changes were made that have not been saved to the data store then this function will save those changes.

Syntax: EnableSave()

Parameters: None

Return type: Boolean indicating success or failure.

GetUserByID

Description: Gets the WMUser object with the ID passed in. You must be logged in for this method to work.

Syntax: GetUserByID(lID as long)

Parameters: lID is a Long Integer specifying the ID of the user to be returned.

Return type: WMUser object with the ID passed in. Returns nothing if the user cannot be found with the given ID.

GetVirtualMailboxByID

Description: Gets the virtual mailbox object with the ID passed in. You must be logged in for this method to work.

Syntax: GetVirtualMailboxByID(lID as long)

Parameters: lID is the Long Integer specifying the ID of the virtual mailbox to be returned.

Return type: WMVirtualMailbox object with the ID passed in. Returns nothing if the virtual mailbox cannot be found with the given ID.

GetISPByID

Description: Gets the ISP object with the ID passed in. You must be logged in for this method to work.

Syntax: GetISPByID(lID as long)

Parameters: lID is the Long Integer specifying the ID of the ISP to be returned.

Return type: WMISP object with the ID passed in. Returns nothing if the ISP cannot be found with the given ID.

GetSentMessages

Description: Gets a list object which contains a snapshot of all sent messages at the time of the method call.

Syntax: GetSentMessages()

Parameters: None

Return type: WMMessageList object containing a snapshot of the sent messages.

GetReceivedMessages

Description: Gets a list object which contains a snapshot of all received messages at the time of the method call.

Syntax: GetReceivedMessages()

Parameters: None

Return type: WMMessageList object contains a snapshot of the received messages.

NewMessage

Description: Create a new WMComposeMessage object to send a mail message.

Syntax: NewMessage()

Parameters: None

Return type: WMComposeMessage object used to send a message

AddGroup

Description: Create a new user group object with the passed in name.

Syntax: AddGroup(sName)

Parameters: sName is a string specifying the name of the new group

Return type: WMGroup object

GetFirstGroup

Description: Gets the first group in the list of WorkgroupMail groups. You must be logged in for this method to work.

Syntax: GetFirstGroup()

Parameters: None

Return type: WMGroup object of the first group. Returns nothing if not logged into the session.

GetNextGroup

Description: Get the next group in the list of WorkgroupMail groups after the group passed into this method. Returns nothing if the group passed in is nothing or is the last group. You must be logged in for this method to work.

Syntax: GetNextGroup (group as object)

Parameters: group is a WMGroup object returned by a previous call to GetFirstGroup() or GetNextGroup().

Return type: WMGroup object of the next group. Returns nothing if there is no next group.

DeleteGroup

Description: Delete the given group from the list of WorkgroupMail groups. You must be logged in for this method to work.

Syntax: DeleteGroup(group as object)

Parameters: group is a WMGroup object returned by a previous method call.

Return type: Boolean indicating success or failure.

LoggedIn

Description: Returns the current WorkgroupMail login state as determined by Login and Logout functions

Syntax: LoggedIn()

Parameters: None

Return type: Boolean indicating true or false.

GetLastErrorString

Description: Returns the last error string returned produced by a function call

Syntax: GetLastErrorString()

Parameters: None

Return type: String which is the last error string text.

GetMessageFromFilename

Description: Returns a message object from a message file. The message file must contain a MIME formatted message.

Syntax: GetMessageFromFilename(sFilename as string)

Parameters: sFilename is a string specifying the message filename

Return type: WMMessage object

LoginUser

Description: Logs into a WorkgroupMail user account

Syntax: Login(sUserAccount sUserPwd)

Parameters: sUserAccount and sUserPwd are string variables specifying the users' account name and password

Return type: WMUser object.

CreateSubFolder

Description: Creates a sub folder under the WorkgroupMail installation folder. Nested sub-folders can be created by use of the backslash character.

Syntax: CreateSubFolder(sFolderName)

Parameters: sFolderName is a string variable representing the sub folders name

Return type: Boolean indicating success or failure.

Example: CreateSubFolder("Logs\Archive").

This will create the folder Archive under the folder Logs. If Logs does not exist then this will be created also.

DeleteSubFolder

Description: Deletes a sub folder under the WorkgroupMail installation folder. Nested sub-folders can be deleted by use of the backslash character.

Syntax: DeleteSubFolder(sFolderName)

Parameters: sFolderName is a string variable specifying the name of the sub folder

Return type: Boolean indicating success or failure.

Example: DeleteSubFolder("Logs\Archive")

This will Delete the folder Archive under the folder Logs

AddPublicFolder

Description: Creates a public message folder

Syntax: AddPublicFolder(sFolderName)

Parameters: sFolderName is a string variable specifying the name of the public folder

Return type: Object of type WMPublicFolder.

Example:

Set PubFold = session. AddPublicFolder(“SalesOrders”)

GetFirstPublicFolder

Description: Returns the details of the first listed public folder

Syntax: GetFirstPublicFolder()

Parameters: None

Return type: Object of type WMPublicFolder.

GetNextPublicFolder

Description: Returns the details of the next listed public folder

Syntax: GetNextPublicFolder(PublicFolder as object)

Parameters: PublicFolder is an object of type WMPublicFolder. It represents the public folder preceding the one required in the public folder list.

Return type: Object of type WMPublicFolder.

DeletePublicFolder

Description: Deletes the specified Public folder

Syntax: DeletePublicFolder(PublicFolder)

Parameters: PublicFolder is an object of type WMPublicFolder. It represents the public folder to be deleted.

Return type: Boolean indicating success or failure.

GetPublicFolderByID

Description: Returns details of the public folder specified by its public folder ID

Syntax: GetPublicFolderByID(ID as long)

Parameters: ID is a long integer specifying the ID

Return type: Object of type WMPublicFolder.

GetGroupByID

Description: Returns details of the group specified by its group ID.

Syntax: GetGroupBy(ID as long)

Parameters: ID is a long integer representing the ID

Return type: Object of type WMGroup.

AddQuarantineArea

Description: Adds a new quarantine area.

Syntax: AddQuarantineArea(sQuarantine as string)

Parameters: sQuarantine is a string specifying the name of the new

quarantine area

Return type: Object of type WMQuarantineArea.

GetFirstQuarantineArea

Description: Returns the details of the first listed quarantine area.

Syntax: GetFirstQuarantineArea()

Parameters: None

Return type: Object of type WMQuarantineArea.

GetNextQuarantineArea

Description: Returns details of the next listed quarantine area.

Syntax: GetNextQuarantineArea(QuarantineArea as object)

Parameters: QuarantineArea is an object of type WMQuarantineArea and is the quarantine area listed before the one required.

Return type: Object of type WMQuarantineArea.

DeleteQuarantineArea

Description: Deletes the specified quarantine area.

Syntax: DeleteQuarantineArea(QuarantineArea as object)

Parameters: QuarantineArea is an object of type WMQuarantineArea and is the quarantine area to be deleted.

Return type: Boolean indicating success or failure.

GetQuarantineAreaByID

Description: Returns details of the quarantine area as specified by its quarantine area ID.

Syntax: GetQuarantineAreaByID(ID as long)

Parameters: ID is a long integer specifying the ID of the quarantine area required.

Return type: Object of type WMQuarantineArea.

AddTrustedHostDomain

Description: Adds a new Trusted Host domain

Syntax: AddTrustedHostDomain(sDomain as string)

Parameters: sDomain is a string specifying the domain name of the new Trusted Host

Return type: Object of type WMAddressOrRange .

AddTrustedHostSingle

Description: Adds a new Trusted Host by IP address

Syntax: AddTrustedHostSingle(lIP as long)

Parameters: lIP is a long integer representing the IP address of the new Trusted Host. The long integer consists of four bytes. The high order byte represents the left hand side of the IP address in dot notation and the low order byte represents the right most part of the IP address

Return type: Object of type WMAddressOrRange.

Example: The IP address 192.168.0.25

could be represented by

C0 A8 00 19 Hex or

3232235545 in long integer (decimal) format

AddTrustedHostRange

Description: Adds a new range of Trusted Hosts by IP address

Syntax: AddTrustedHostRange(lIPstart as long lIPend as long)

Parameters: lIPstart and lIPend are long integers specifying the start and end IP address of the new Trusted Host range. Each long integer consists of four bytes. The high order byte represents the left hand side of the IP address in dot notation and the low order byte represents the right most part of the Ip address

Return type: Object of type WMAddressOrRange.

Example: The IP address 192.168.0.25

could be represented by

C0 A8 00 19 Hex or

3232235545 in long integer (decimal) format

DeleteTrustedHost

Description: Deletes a Trusted Host

Syntax: DeleteTrustedHost(Host as object)

Parameters: Host is the object of type WMAddressOrRange.

Return type: Boolean indicating success or failure.

GetFirstTrustedHost

Description: Returns the first Trusted Host listed in WorkgroupMail

Syntax: GetFirstTrustedHost()

Parameters: None

Return type: Object of type WMAddressOrRange.

GetNextTrustedHost

Description: Returns the next Trusted Host listed in WorkgroupMail

Syntax: GetNextTrustedHost(TrustedHost as object)

Parameters: TrustedHost is an object of type WMAddressOrRange and is the Trusted Domain listed before the one required.

Return type: Object of type WMAddressOrRange.

AddDomain

Description: Adds a new Domain

Syntax: AddDomain(sDomain as string)

Parameters: sDomain is a string specifying the name of the new domain

Return type: Object of type WMAddressOrRange.

GetFirstDomain

Description: Returns the details of the first listed Domain

Syntax: GetFirstDomain()

Parameters: None

Return type: Object of type WMAddressOrRange.

GetNextDomain

Description: Returns the details of the next listed Domain

Syntax: GetFirstDomain(Domain as object)

Parameters: Domain is an object of type WMAddressOrRange and is the Domain listed before the one required.

Return type: Object of type WMAddressOrRange.

DeleteDomain

Description: Deletes the specified Domain

Syntax: DeleteDomain()

Parameters: Object of type WMAddressOrRange

Return type: Boolean indicating success or failure.

FindTrustedHostByDescription

Description: Identifies a Trusted Host by a search string

Syntax: FindTrustedHostByDescription(sHost as string)

Parameters: sHost is a string used to identify the Trusted Host

Return type: Object of type WMAddressOrRange

ARGetFirstSendException

Description: Returns the first ‘send exception’ listed in the default account settings for when a user is not allowed to send mail to the outside world, except to …

Syntax: ARGetFirstSendException()

Parameters: None

Return type: String specifying the first send exception e-mail address.

ARGetNextSendException

Description: Returns the next ‘send exception’ listed in the default account settings for when a user is not allowed to send mail to the outside world, except to …

Syntax: ARGetNextSendException(sEmailAddr as string)

Parameters: sEmailAddr is a string variable specifying the current send exception e-mail address

Return type: String specifying the next send exception e-mail address.

ARGetFirstReceiveException

Description: Returns the first ‘receive exception’ listed in the default account settings for when a user is not allowed to receive mail from the outside world, except from …

Syntax: ARGetFirstReceiveException()

Parameters: None

Return type: String specifying the first receive exception e-mail address.

ARGetNextReceiveException

Description: Returns the next ‘receive exception’ listed in the default account settings for when a user is not allowed to receive mail from the outside world, except from …

Syntax: ARGetNextReceiveException(sEmailAddr as string)

Parameters: sEmailAddr is a string variable specifying the current receive exception e-mail address

Return type: String specifying the next receive exception e-mail address.

ARAddSendException

Description: Adds a new Send exception address to the list

Syntax: ARAddSendException(sSendAddr as string)

Parameters: sSendAddr is a string variable specifying the new send exception e-mail address.

Return type: Boolean indicating success or failure.

ARAddReceiveException

Description: Adds a new Receive exception address to the list

Syntax: ARAddReceiveException(sReceiveAddr as string)

Parameters: sReceiveAddr is a string variable specifying the new

receive exception e-mail address.

Return type: Boolean indicating success or failure.

ARDeleteSendException

Description: Deletes the Send exception address specified in the string parameter

Syntax: ARDeleteSendException(sSendAddr as string)

Parameters: sSendAddr is a string variable specifying the send exception address to be deleted

Return type: Boolean indicating success or failure.

ARDeleteReceiveException

Description: Deletes the Receive exception address specified in the string parameter

Syntax: ARDeleteReceiveException(sReceiveAddr as string)

Parameters: sReceiveAddr is a string variable specifying the receive exception address to be deleted

Return type: Boolean indicating success or failure.

AddSpamServer

Description: Adds a new Spam Server

Syntax: AddSpamServer (sSpamServer as string)

Parameters: sSpamServer is a string specifying the name of the new spam server to be listed

Return type: Object of type WMSpamServer.

GetFirstSpamServer

Description: Returns the details of the first listed Spam Server

Syntax: GetFirstSpamServer()

Parameters: None

Return type: Object of type WMSpamServer.

GetNextSpamServer

Description: Returns details of the next listed Spam Server

Syntax: GetNextSpamServer(SpamServer as object)

Parameters: SpamServer is an object of type WMSpamServer which specifies the current Spam Server

Return type: Object of type WMSpamServer which specifies the next listed Spam Server

DeleteSpamServer

Description: Deletes the specified Spam Server

Syntax: DeleteSpamServer(SpamServer as object)

Parameters: SpamServer is an object of type WMSpamServer which represents the Spam Server to be deleted.

Return type: Boolean indicating success or failure.

GetSpamServerByID

Description: Returns details of the Spam Server as specified by its Spam Server ID

Syntax: GetSpamServerByID(ID as long)

Parameters: ID is a long integer representing the ID of the required Spam Server

Return type: Object of type WMSpamServer.

WMDomain_

The WMDomain_ object provides properties and methods to interrogate and edit domain details. Various methods and properties of the WMSession object and other objects return a WMDomain_ object and take WMDomain_ objects as arguments.

More:

Properties

Methods

Properties

UnKnownRedirectUser

Description: The user account that receives all messages for unknown users that are received for this domain.

Type: Object of type WMUser

RejectUnknown

Description: Specifies whether or not mail for unknown recipients is rejected when receiving via SMTP.

Type: Boolean

RelayUnknown

Description: Specifies if mail for unknown users can be relayed. If true and the recipient address has an entry in the routing table then the message will be relayed.

Type: Boolean

SendISP

Description: If the send method for this domain is via an ISP then this property should be configured with the details of the sending ISP. If the send method is 'Direct' then this item will contain nothing.

Type: Object of type WMISP

RelayUnknownAlways

Description: Sets or gets whether messages sent to an unknown mailbox at the domain will be relayed (redirected) to the Outgoing queue. This corresponds to the user interface in the Unknown Recipients page of the Domain property sheet.

Type: Object of type WMISP

Methods

AddDomain

Description: Creates a domain name entry

Syntax: AddDomain (sDomain as string)

Parameters: sDomain is a string representing the new domain name.

Return type: Boolean indicating success or failure.

GetFirstDomain

Description: Returns the first listed domain

Syntax: GetFirstDomain ()

Parameters: None.

Return type: String specifying the name of the domain

GetNextDomain

Description: Returns the next listed domain

Syntax: GetNextDomain (sDomain as string)

Parameters: sDomain is a string representing the current domain name.

Return type: String specifying the name of the next domain

DeleteDomain

Description: Deletes the specified domain

Syntax: DeleteDomain (sDomain as string)

Parameters: sDomain is a string representing the domain to be deleted.

Return type: Boolean indicating success or failure.

GetID

Description: Returns the ID of the current domain

Syntax: GetID ()

Parameters: None

Return type: Long integer specifying the ID of the current domain

GetSendDirect

Description: Returns flag specifying if the domain is configured to send direct

Syntax: GetSendDirect()

Parameters: None

Return type: Boolean indicating if send direct is selected

SetSendDirect

Description:	Sets the flag which specifies that the domain is configured to send direct
Syntax:	SetSendDirect()
Parameters:	None
Return type:	None

WMGroup

The WMGroup object encapsulates a WorkgroupMail local user and provides properties and methods to interrogate and edit user details. Various methods and properties of the WMSession object and other objects return a WMGroup object and take WMGroup objects as arguments.

More:

Properties

Methods

Properties

Name

Description: The name of the group.

Type: String

Methods

AddUser

Description:	Adds a user to the listing of a group
Syntax:	AddUser (Member as object)
Parameters:	Member is an object of type WMUser.
Return type:	Boolean indicating success or failure

GetFirstUser

Description:	Returns the first user listed in the group
Syntax:	GetFirstUser ()
Parameters:	None.
Return type:	Object of type WMUser specifying the first user listed in the group

GetNextUser

Description:	Returns the next user listed in the group
Syntax:	GetNextUser (Member as object)
Parameters:	Member is an object of type WMUser.
Return type:	Object of type WMUser specifying the next user listed in the group

RemoveUser

Description:	Removes a user from the listing of a group
Syntax:	RemoveUser (Member as object)
Parameters:	Member is an object of type WMUser.
Return type:	Boolean indicating success or failure

GetID

Description:	Adds a user to the listing of a group
Syntax:	GetID ()
Parameters:	None.
Return type:	Long integer specifying the ID of the current WorkgroupMail Group

WMISP

The WMISP object encapsulates a WorkgroupMail ISP and provides properties and methods to interrogate and edit ISPs details. Various methods and properties of the WMSession object and other objects return a WMISP object and take WMISP objects as arguments.

More:

Properties

Methods

Properties

Name

Description: This property allows the user to get and set the name of the WMISP object. This is the name that is displayed in the WorkgroupMail administrator.

Type: String

Enabled

Description: The Enabled property allows the user to get and set if a WMISP object is enabled or disabled.

Type: Boolean

SMTPReceive

Description: Allows the user to get and set whether this ISP receives mail via SMTP.

Type: Boolean indicating if SMTP receive is enabled

SendServerAddress

Description: Allows the user to get and set the address of the ISPs send (SMTP) server.

Type: String. Eg “216.54.12.9” or “mail.isp.net”

ReceiveServerAddress

Description: Allows the user to get and set the address of the ISP's receive (POP3) server.

Type: String. Eg “216.54.12.9” or “mail.isp.net”

UseModem

Description: Allows the user to get and set if this ISP is connected to the internet via a modem (as opposed to a permanent connection).

Type: Boolean

DialUpService

Description: Allows the user to get and set the dial-up service for the ISP if it is connected via a modem.

Type: String equating to the name of the dial up service.

ConnectOnStartup

Description: Allows the user to get and set if this ISP is makes a connection on startup of the WorkgroupMail service.

Type: Boolean.

StartupConnectDelay

Description: Specifies the number of seconds to wait before connecting to the ISP on startup.

Type: Long Integer specifying the number of seconds' delay.

ConnectWhenQueueFull

Description: Allows the user to get and set if the WorkgroupMail service is to connect to this ISP when there are a certain number of sent messages pending. The number of messages pending is defined in QueueFullSize

Type: Boolean.

QueueFullSize

Description: Specifies the maximum number of messages allowed in the Sent Messages queue before the WorkgroupMail service will connect to the ISP.

Type: Long Integer representing the maximum number of messages..

ConnectWhenMessagesWaiting

Description: Allows the user to get and set if the WorkgroupMail service is to connect to this ISP if sent messages have been waiting for a set period of time. The waiting time for messages is defined in MessageWaitTime

Type: Boolean.

MessageWaitTime

Description: Specifies the maximum time a message will wait before the WorkgroupMail service will connect to the ISP.

Type: Long Integer specifying the time in minutes

ConnectFrequently

Description: Allows the user to get and set whether the WorkgroupMail service connects to the ISP on a schedule. The frequency of the connection is defined in ConnectFrequency.

Type: Boolean

ConnectFrequency

Description: Specifies the scheduled connection interval.

Type: Long Integer representing interval in minutes

POP3DetailsActive

Description: Allows the user to get and set whether this ISP receives via POP3.

Type: Boolean

POP3LoginName

Description: Allows the user to get and set the multiple user POP3 account username for this ISP

Type: String

POP3Password

Description: Allows the user to get and set the multiple user POP3 account password for this ISP

Type: String

LeaveMessagesOnServer

Description: Allows the user to get and set if downloaded message are to be left on the server for this ISP.

Type: Boolean

DeleteOldMessagesFromServer

Description: Allows the user to get and set if old messages are to be deleted from the server.

Type: Boolean

ServerDeleteMessageAge

Description: Allows the user to get and set the age at which old messages on the server are deleted, in days.

Type: Long Integer

UnknownRedirectUser

Description: Specifies the user account that will receive messages that don't match the e-mail address of any user mailbox

Type: Object of type WMUser

OwnsDomains

Description: This method is not valid for version 7. It applies to version 6 only.

Type: Boolean

Methods

AddDomain

Description: Adds a domain to the ISP.

Syntax: AddDomain(string sDomain)

Parameters: sDomain is the name of the domain to be added.

Return type: Boolean indicating success or failure.

GetFirstDomain

Description: Gets the first domain listed for the ISP.

Syntax: GetFirstDomain()

Parameters: None

Return type: String containing the first domain of the ISP.

GetNextDomain

Description: Gets the next domain listed for the ISP

Syntax: GetNextDomain(sDomain as string)

Parameters: sDomain is a string specifying the current domain as returned from a previous call to GetFirstDomain() or GetNextDomain().

Return type: String containing the next domain of the ISP.

DeleteDomain

Description: Deletes the specified domain from the ISP.

Syntax: DeleteDomain(sDomain as string)

Parameters: sDomain is the domain to be deleted.

Return type: Boolean indicating success or failure.

GetConnectionSchedule

Description: Gets the connection schedule for the ISP.

Syntax: GetConnectionSchedule()

Parameters: None

Return type: Object of type WMSchedule representing the connection schedule for the ISP.

GetID

Description: Get the ID of the current ISP, a number which uniquely identifies the ISP.

Syntax: GetID()

Parameters: None

Return type: Long Integer.

GetFirstProperty

Description:	Get the first property from the list of properties for the current ISP.
Syntax:	GetFirstProperty()
Parameters:	None
Return type:	Object of type WMProperty.

GetNextProperty

Description:	Get the next property from the list of properties for the current ISP.
Syntax:	GetNextProperty(property as object)
Parameters:	property is a WMProperty object specifying the current property.
Return type:	WMProperty object containing the next property.

WMUser

The WMUser object encapsulates a WorkgroupMail local user and provides properties and methods to interrogate and edit user details. Various methods and properties of the WMSession object and other objects return a WMUser object and take WMUser objects as arguments.

More:

Properties

Methods

Properties

Name

Description: This property allows the user to get and set the name of the WMUser object.

Type: String

Enabled

Description: The Enabled property allows the user to get and set if a WMUser object is enabled or disabled.

Type: Boolean

LocalPOP3LoginName

Description: Gets or sets the local account name for this user. The local account name forms part of the credentials for logging a user onto the local POP3 server, the local IMAP server or WebMail.

Type: String

LocalPOP3Password

Description: Gets or sets the local account password for this user. The local account password forms part of the credentials for logging a user onto the local POP3 server, the local IMAP server or WebMail.

Type: String

UseDefaultQuotaSettings

Description: Gets or sets whether or not this user will use or override the default account quota settings that are globally defined.

Type: Boolean

UseDefaultAcntPruningSettings

Description: Gets or sets whether or not this user will use or override the default account pruning settings that are globally defined.

Type: Boolean

UseDefaultRestrictionSettings

Description: Gets or sets whether or not this user will use or override the default account restriction settings that are globally defined.

Type: Boolean

AQLimitStoredMessages

Description: A boolean that specifies if a limit will be imposed on the maximum number of messages allowed this user's account mailbox. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Quotas page of the User property sheet.

Type: Boolean

AQStoredMessageLimit

Description: The account quota for the maximum number of messages that can be stored in this user's account mailbox. Dependent on AQLimtStoredMessages being true. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Quotas page of the User property sheet.

Type: Long Integer

AQLimitStorageSize

Description: A boolean setting that specifies if a limit will be imposed on the maximum amount of disk space that can be occupied by messages in this user's account mailbox. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Quotas page of the User property sheet.

Type: Boolean

AQStorageSizeLimit

Description: The account quota, in Kb, for the maximum amount of disk space that can be occupied by messages in this user's account mailbox. Dependent on AQLimtStorageSize being true. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Quotas page of the User property sheet.

Type: Long Integer

APAutoDeleteInactiveAccount

Description: A boolean setting that specifies if inactive accounts will be deleted automatically. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the User

property sheet.

Type: Boolean

APInactiveDays

Description: The number of days that an account has to be inactive before it is automatically deleted . Dependent on APAutoDeleteInactiveAccount being true. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the User property sheet.

Type: Long Integer

APDeleteInboxMessages

Description: A boolean setting that specifies if old messages in this user's inbox will be purged. The Inbox is where WorkgroupMail initially delivers new messages for a particular user. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the User property sheet.

Type: Boolean

APDeleteInboxMessageDays

Description: The default number of days old that a message must be before it is automatically purged from this user's Inbox. Dependent on APDeleteInboxMessages being true. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the User property sheet.

Type: Long Integer

APDeletePersonalFolderMessages

Description: A boolean setting that specifies if old messages in any of this user's personal folders will be purged. Personal folders are folders that are created in WebMail or using an IMAP client. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the User property sheet.

Type: Boolean

APDeletePFMessageDays

Description: The number of days old that a message must be before it is automatically purged from any of this user's personal folders. Dependent on APDeleteInboxMessages being true. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the User property sheet.

Type: Long Integer

APDeleteDeletedMessages

Description: A boolean setting that specifies if deleted messages will be purged for this user. Deleted messages are messages that exist in the Deleted Items folder. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the User property sheet.

Type: Boolean

APDeleteDeletedMessageDays

Description: The number of days old that a deleted message must be before it is automatically purged from this user's account. Dependent on APDeleteDeletedMessages being true. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Pruning page of the User property sheet.

Type: Long Integer

ARCanUseWebMail

Description: A boolean setting that specifies whether or not this user can use WebMail. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Restrictions page of the User property sheet.

Type: Boolean

ARCanUseIMAP

Description: A boolean setting that specifies whether or not this user can use IMAP. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Restrictions page of the Settings property sheet.

Type: Boolean

ARCanSendExternalMessages

Description: A boolean setting that specifies whether or not this user can send external mail. If false, the exceptions list will prevail. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Restrictions page of the User property sheet.

Type: Boolean

ARCanReceiveExternalMessages

Description: A boolean setting that specifies whether or not this user can receive external mail. If false, the exceptions list will prevail. This setting overrides the global equivalent setting. This setting only applies to the Enterprise edition. It represents the corresponding user interface settings in the Account Restrictions page of the Settings property sheet.

Type: Boolean

Methods

SetNoForwarding

Description: Sets the user to have no forwarding.

Syntax: SetNoForwarding()

Parameters: None

Return type: Boolean indicating success or failure.

SetForwardUser

Description: Sets the WMUser object that is the user to forward messages to, sent to this user.

Syntax: SetForwardUser(user as object)

Parameters: user is a WMUser object. It must not be the same as this user object.

Return type: Boolean indicating success or failure.

GetForwardUser

Description: Gets the WMUser object that is the user that messages sent to this user are forwarded to.

Syntax: GetForwardUser()

Parameters: None

Return type: WMUser object of the forward user.

GetISPSettings

Description: Get the WMUserISPSettings object for this user which contains all of this user's settings for a specified ISP.

Syntax: GetISPSettings(isp as object)

Parameters: isp is a WMISP object for which is the settings are to be got.

Return type: WMUserISPSettings object containing the settings for the given ISP for this user.

GetID

Description: Get the ID of this user which is a number which uniquely identifies this user.

Syntax: GetID()

Parameters: None

Return type: Long Integer integer containing the ID of this user.

GetFirstProperty

Description: Get the first property from the list of properties for this user.

Syntax: GetFirstProperty()

Parameters: None

Return type: WMProperty object containing the first property of this user.

GetNextProperty

Description: Get the next property on from the property passed into this function.

Syntax: GetNextProperty(property as object)

Parameters: property is a WMProperty object. The next property after this one is returned.

Return type: WMProperty object containing the next property.

GetAddresses

Description: Gets a string of all the valid addresses for the user.

Syntax: GetAddresses()

Parameters: None

Return type: String containing all the valid addresses for this user.

SetForwardGroup

Description: Sets the forwarding group for this mailbox to be the group object supplied

Syntax: SetForwardGroup(group as object)

Parameters: group is an object of type WMGroup.

Return type: Boolean indicating success or failure

GetForwardGroup

Description: Gets the forwarding group for this mailbox

Syntax: GetForwardGroup()

Parameters: None.

Return type: WMGroup object representing the forwarding group

GetSendAddress

Description: Returns the forwarding e-mail address specified for this mailbox.

Syntax: GetSendAddress()

Parameters: None.

Return type: String of form **Error! Bookmark not defined.**

GetDomainSettings

Description: Gets the domain settings object for this mailbox

Syntax: GetDomainSettings(domain as object)

Parameters: WMDomain

Return type: WMDomain object

AddDomainSetting

Description: Adds a domain settings object for this mailbox.

Syntax: AddDomainSettings(domain as object, sMailbox as string)

Parameters: domain is an object of type WMDomain

sMailbox is a string

Return type: UserDomainSettings object

DeleteDomainSetting

Description: Deletes the domain setting object for this mailbox

Syntax: DeleteDomainSettings(settings)

Parameters: settings object.

Return type: Boolean indicating success or failure

AddUserDefinedProperty

Description: Adds a user defined property with a name of sName and gives it a value of sValue.

Syntax: AddUserDefinedProperty(sName, sValue)

Parameters: sName is a string which represents the name of the user defined property.

sValue is a string which represents the value of the user defined property.

Return type: An object which represents the new WMProperty object that was added to the user.

DeleteUserDefinedProperty

Description: Deletes the passed in user defined property

Syntax: AddUserDefinedProperty(property)

Parameters: property is the WMProperty object that will be deleted.

Return type: A boolean that returns true if the property was successfully deleted.

SetForwardAddress

Description: Sets the e-mail address that messages sent to this user will be automatically forwarded to. This corresponds with the user interface in the Forwarding page of the User property sheet.

Syntax: SetForwardAddress(sAddress)

Parameters: sAddress is the e-mail address that messages should be

forwarded to.

Return type: A boolean that returns true if the forward address was successfully configured.

GetForwardAddress

Description: Gets the e-mail address that messages sent to this user will be automatically forwarded to. This corresponds with the user interface in the Forwarding page of the User property sheet.

Syntax: GetForwardAddress()

Parameters: None.

Return type: A string that represents the e-mail address to which mail for this user is automatically forwarded.

GetReceivedMessages

Description: Returns a WMMessageList object, which itself contains the list of messages in this user's InBox.

Syntax: GetReceivedMessages()

Parameters: None.

Return type: A WMMessageList object.

AddFolder

Description: Adds a personal folder to this user's list of folders.

Syntax: AddFolder(sFolderName)

Parameters: sFolderName is a string that represents the name of the personal folder.

Return type: A WMFolder object that represents the folder that was added.

GetFirstFolder

Description: Gets the first folder belonging to this user.

Syntax: GetFirstFolder()

Parameters: None.

Return type: A WMFolder object that represents the first folder belonging to this user.

GetNextFolder

Description: Gets the next folder belonging to this user.

Syntax: GetNextFolder(folder)

Parameters: The folder returned by a previous call to GetFirstFolder or GetNextFolder.

Return type: A WMFolder object that represents the next folder in the list belonging to this user.

GetFolderFromName

Description: Gets the folder belonging to this user that matches the passed in name.

Syntax: GetFolderFromName(sName)

Parameters: sName is a string.

Return type: A WMFolder object that matches the passed in folder name.

ARGetFirstSendException

Description: Returns the first ‘send exception’ listed in this user’s account restrictions for when this user is not allowed to send mail to the outside world, except to …

Syntax: ARGetFirstSendException()

Parameters: None

Return type: String specifying the first send exception e-mail address.

ARGetNextSendException

Description: Returns the next ‘send exception’ listed in this user’s account restrictions for when this user is not allowed to send mail to the outside world, except to …

Syntax: ARGetNextSendException(sEmailAddr as string)

Parameters: sEmailAddr is a string variable specifying the current send exception e-mail address

Return type: String specifying the next send exception e-mail address.

ARGetFirstReceiveException

Description: Returns the first ‘receive exception’ listed in this user’s account restrictions for when this user is not allowed to receive mail from the outside world, except from …

Syntax: ARGetFirstReceiveException()

Parameters: None

Return type: String specifying the first receive exception e-mail address.

ARGetNextReceiveException

Description: Returns the next ‘receive exception’ listed in this user’s account restrictions for when this user is not allowed to receive mail from the outside world, except from …

Syntax: ARGetNextReceiveException(sEmailAddr as string)

Parameters: sEmailAddr is a string variable specifying the current receive exception e-mail address

Return type: String specifying the next receive exception e-mail

address.

ARAddSendException

Description: Adds a new Send exception address to the list

Syntax: ARAddSendException(sSendAddr as string)

Parameters: sSendAddr is a string variable specifying the new send exception e-mail address.

Return type: Boolean indicating success or failure.

ARAddReceiveException

Description: Adds a new Receive exception address to the list

Syntax: ARAddReceiveException(sReceiveAddr as string)

Parameters: sReceiveAddr is a string variable specifying the new receive exception e-mail address.

Return type: Boolean indicating success or failure.

ARDeleteSendException

Description: Deletes the Send exception address specified in the string parameter

Syntax: ARDeleteSendException(sSendAddr as string)

Parameters: sSendAddr is a string variable specifying the send exception address to be deleted

Return type: Boolean indicating success or failure.

ARDeleteReceiveException

Description: Deletes the Receive exception address specified in the string parameter

Syntax: ARDeleteReceiveException(sReceiveAddr as string)

Parameters: sReceiveAddr is a string variable specifying the receive exception address to be deleted

Return type: Boolean indicating success or failure.

WMVirtualMailbox

The WMVirtualMailbox object encapsulates a WorkgroupMail virtual mailbox and provides properties and methods to interrogate and edit virtual mailbox details. Various methods and properties of the WMSession object and other objects return a WMVirtualMailbox object and take WMVirtualMailbox objects as arguments.

More:

Properties

Methods

Properties

Name

Description: This property allows the user to get and set the name of the WMVirtualMailbox object.

Type: String

Enabled

Description: The Enabled property allows the user to get and set if a WMVirtualMailbox object is enabled or disabled.

Type: Boolean

AutoResponder

Description: Gets or sets whether this virtual mailbox is an auto responder.

Type: Boolean

ResponderFile

Description: A string property which represents the filename of the file, which contains the auto responder text. This is a read/write property.

Type: String

TextFormatResponder

Description: A boolean read/write property which is true if the ResponderFile is a text-formatted response file. If this property is false then WorkgroupMail expects the ResponderFile to be HTML formatted.

Type: String

ResponderAttachment

Description: A string read/write property which represents the filename of the file (if any), which should be attached to the response message.

Type: String

ResponderReplyAddress

Description: A string read/write property which represents the reply address of response messages sent out by this auto responder. This corresponds to the user interface in the Auto-Responding page of the Auto Responder property sheet.

Type: String

Methods

SetNoForwarding

Description: Sets the virtual mailbox to have no forwarding.

Syntax: SetNoForwarding()

Parameters: None

Return type: Boolean indicating success or failure.

SetForwardUser

Description: Sets the WMUser object that is the user to forward messages to, sent to this virtual mailbox.

Syntax: SetForwardUser(user as object)

Parameters: user is a WMUser object.

Return type: Boolean indicating success or failure.

GetForwardUser

Description: Gets the WMUser object that is the user that messages sent to this virtual mailbox are forwarded to.

Syntax: GetForwardUser()

Parameters: None

Return type: WMUser object of the forward user.

GetISPSettings

Description: Get the WMUserISPSettings object for this virtual mailbox which contains all of this virtual mailbox's settings for a specified ISP.

Syntax: GetISPSettings(isp as object)

Parameters: isp is a WMISP object for which is the settings are to be got.

Return type: WMUserISPSettings object containing the settings for the given ISP for this virtual mailbox.

GetID

Description: Get the ID of this virtual mailbox which is a number which uniquely identifies this virtual mailbox.

Syntax: GetID()

Parameters: None

Return type: Long Integer integer containing the ID of this virtual mailbox.

GetFirstProperty

Description: Get the first property from the list of properties for this

virtual mailbox.

Syntax: GetFirstProperty()

Parameters: None

Return type: WMProperty object containing the first property of this virtual mailbox.

GetNextProperty

Description: Get the next property on from the property passed into this function.

Syntax: GetNextProperty(property as object)

Parameters: property is a WMProperty object. The next property after this one is returned.

Return type: WMProperty object containing the next property.

GetAddresses

Description: Gets a string of all the valid addresses for the virtual mailbox.

Syntax: GetAddresses()

Parameters: None

Return type: String containing all the valid addresses for this virtual mailbox.

GetDomainSettings

Description: Gets the domain settings object for this mailbox

Syntax: GetDomainSettings(domain as object)

Parameters: WMDomain

Return type: WMDomain object

AddDomainSetting

Description: Adds a domain settings object for this mailbox.

Syntax: AddDomainSettings(domain as object, sMailbox as string)

Parameters: domain is an object of type WMDomain

sMailbox is a string

Return type: UserDomainSettings object

DeleteDomainSetting

Description: Deletes the domain setting object for this mailbox

Syntax: DeleteDomainSettings(settings)

Parameters: settings object.

Return type: Boolean indicating success or failure

AddUserDefinedProperty

Description: Adds a user defined property with a name of sName and gives it a value of sValue.

Syntax: AddUserDefinedProperty(sName, sValue)

Parameters: sName is a string which represents the name of the user defined property.

sValue is a string which represents the value of the user defined property.

Return type: An object which represents the new WMProperty object that was added to the virtual mailbox.

DeleteUserDefinedProperty

Description: Deletes the passed in user defined property

Syntax: AddUserDefinedProperty(property)

Parameters: property is the WMProperty object that will be deleted.

Return type: A boolean that returns true if the property was successfully deleted.

WMUserISPSettings

The WMUserISPSettings encapsulates the object which maintains the settings for a user in relation to one ISP. This object can be accessed via a call to GetISPSettings() on the WMUser or WMVirtualMailbox object.

More:

Properties

Methods

Properties

Active

Description: Allows the user to get or set if these user ISP settings are active.

Type: Boolean

POP3LoginName

Description: The POP3 login name for the associated user's personal POP account with this ISP. WorkgroupMail uses this information to log onto the POP account at the ISP in order to download mail for the user from the ISP into WorkgroupMail.

Type: String

POP3Password

Description: The POP3 password for the associated user's personal POP account with this ISP. WorkgroupMail uses this information to log onto the POP account at the ISP in order to download mail for the user from the ISP into WorkgroupMail.

Type: String

Methods

GetID

Description:	Gets the ID of the settings.
Syntax:	GetID()
Parameters:	None
Return type:	A integer that returns the ID of the record which links the associated user with the ISP.

WMUserDomainSettings

The WMUserDomainSettings object encapsulates the object which maintains the settings for a user in relation to a domain. This object can be accessed via a call to GetDomainSettings() on the WMUser or WMVirtualMailbox object.

More:

Properties

Methods

Properties

User

Description: The user that is associated with these settings.

Type: WMUser object.

Domain

Description: The domain that is associated with these settings.

Type: WMDomain object.

Mailbox

Description: Each user in a domain is differentiated by its unique mailbox. For example, the user Fred Smith may be part of the **company.com** domain and may have an associated email address of **fred@company.com**. This user's Mailbox property for the company.com domain would be **fred**.

Type: String

Methods

GetID

Description:	Gets the ID of the settings.
Syntax:	GetID()
Parameters:	None
Return type:	A integer that returns the ID of the record which links the associated user with the domain

WMSchedule

The WMSchedule object encapsulates a schedule of when to connect to an ISP.

More:

Properties

Methods

Properties

CellState

Description: Allows the user to get and set the cell state for the WMSchedule object given an hour and a day. The hour must be between 1 and 24 and the day must be between 1 and 7.

Syntax: boolean CellState(short iHour, short iDay)

void CellState(short iHour, short iDay, long bNewValue)

Example:

```
bCellState = schedule.CellState(17, 2)
            schedule.CellState(17, 2) = bCellState
```

Methods

WMSchedule has no methods.

WMProperty

The WMProperty object encapsulates one name/value pair for the property of any object.

More:

Properties

Methods

Properties

Name

Description: The name of the property. This is a read only property.

Type: String

Value

Description: The value of the property. This is a read only property.

Type: String

Methods

WMProperty has no methods.

WMFolder

The WMUser object encapsulates a WorkgroupMail local user and provides properties and methods to interrogate and edit user details. Various methods and properties of the WMSession object and other objects return a WMUser object and take WMUser objects as arguments.

More:

Properties

Methods

Properties

Name

Description: The name given to this folder.

Type: String

Methods

GetMessages

Description: Returns a list of messages belonging in this folder.

Syntax: GetMessages ()

Return type: WMMessageList

GetFirstSubFolder

Description: Returns the first sub folder belonging to this folder.

Syntax: GetFirstSubFolder ()

Return type: WMFolder

GetNextSubFolder

Description: Returns the next sub folder belonging to this folder.

Syntax: GetFirstSubFolder (folder)

Parameters: folder is the folder returned by a previous call to GetFirstSubFolder() or GetNextSubFolder().

Return type: WMFolder

AddSubFolder

Description: Adds a sub folder with the passed in name to this folder.

Syntax: AddSubFolder (sFolder)

Parameters: sFolder is the name given to the new sub folder

Return type: WMFolder

DeleteSubFolder

Description: Deletes the passed in sub folder and optionally the messages in that folder.

Syntax: DeleteSubFolder (folder, bDeleteMessages)

Parameters: folder is the sub folder that is to be deleted.

BDeleteMessages should be set to True if you want to delete the messages in the sub folder.

Return type: Boolean

GetSubFolderFromName

Description: Gets a sub folder by name.

Syntax: GetSubFolderByName (sFolder)

Parameters: sFolder is the name of the subfolder that you wish to return.

Return type: WMFolder

GetMessageCount

Description: Gets the number of messages in this folder.

Syntax: GetMessageCount()

Parameters: None

Return type: An integer representing the number of messages in the folder.

GetFirstProperty

Description: Get the first property from the list of properties for this folder.

Syntax: GetFirstProperty()

Parameters: None

Return type: WMProperty object containing the first property of this folder.

GetNextProperty

Description: Get the next property on from the property passed into this function.

Syntax: GetNextProperty(property as object)

Parameters: property is a WMProperty object. The next property after this one is returned.

Return type: WMProperty object containing the next property.

WMPublicFolder

The WMPublicFolder object encapsulates a WorkgroupMail public folder.

More:

Properties

Methods

Properties

Name

Description: The display name of the public folder.

Type: String

Rule

Description: The action taken by this public folder on the messages inside.

Type: Integer. Permitted values are: 0 = Keep, 1 = Move messages, 2 = Delete messages.

MoveToFolder

Description: If the Rule property is set to 1 (Move Messages), then this property represents the WMFolder to which the messages should be moved.

Type: WMFolder object.

AfterMinutes

Description: If the Rule property is set to 1 (Move Messages) or 2 (Delete messages), then this property represents the number of minutes that the message must reside in the public folder before having the move or delete rule applied.

Type: Integer.

Methods

GetMessages

Description: Returns the a WMMessageList that contains the list of messages in the public folder.

Syntax: GetMessages()

Parameters: None

Return type: WMMessageList.

GetMessageCount

Description: Returns the number of messages in the quarantine.

Syntax: GetMessageCount()

Parameters: None

Return type: Integer.

WMQuarantineArea

The WMQuarantineArea object encapsulates a WorkgroupMail quarantine area.

More:

Properties

Methods

Properties

Name

Description: The display name of the quarantine.

Type: String

Description

Description: The description shown for the quarantine area in the WorkgroupMail administrator.

Type: String

Rule

Description: The action taken by this public folder on the messages inside.

Type: Integer. Permitted values are: 0 = Keep, 1 = Allow messages through, 2 = Delete messages.

AfterMinutes

Description: If the Rule property is set to 1 (Move Messages) or 2 (Delete messages), then this property represents the number of minutes that the message must reside in the quarantine before having the move or delete rule applied.

Type: Integer.

TimeOption

Description: If the Rule property is set to 1 (Move Messages) or 2 (Delete messages), then this property represents whether the rule will be applied after a certain period of time (according to the AfterMinutes property), or in a certain time range (according to the GetTimeRange method).

Type: Integer. Permitted values are: 0 = After a certain period of time, 1 = In a certain time range.

Methods

GetIncomingMessages

Description: Returns the a WMMessageList that contains the list of incoming messages in the quarantine.

Syntax: GetIncomingMessages()

Parameters: None

Return type: WMMessageList.

GetOutcomingMessages

Description: Returns the a WMMessageList that contains the list of outcoming messages in the quarantine.

Syntax: GetOutcomingMessages()

Parameters: None

Return type: WMMessageList.

GetTimeRange

Description: If the Rule property is set to 1 (Move Messages) or 2 (Delete messages), and the TimeOption is set to 1 (In a certain time range) then this property represents the time range of interest. The returned object is a WMSchedule object.

Syntax: GetTimeRange()

Parameters: None

Return type: WMSchedule object.

ReleaseMessage

Description: Releases the message from the quarantine.

Syntax: ReleaseMessage()

Parameters: None

Return type: Boolean. True if successful, False otherwise.

WMMessage

The WMMessage object encapsulates a sent or received message, allowing the user of the API to determine detains of messages that have been sent by clients but not sent to the appropriate ISP yet and messages that have been received into WorkgroupMail but not collected by clients yet.

More:

Properties

Methods

Properties

Subject

Description: The subject of the message. This is a read only property.

Type: String

Received

Description: Gets the received time of a received message. This is a read only property.

Type: DATE

Sent

Description: Get the sent time of a sent message. This is a read only property.

Type: DATE

Sender

Description: Gets the sender address for the message. This property is read only.

Type: String

Recipients

Description: This property gets a string containing all the recipient addresses. This is a read only property.

Type: String

Size

Description: Gets the size of the message in bytes. This is a read only property.

Type: Long Integer

Account

Description: This property gets the name of the ISP account that this sent message is using. This is a read only property.

Type: String

Filename

Description: This read only property gets the filename that stores the sent or received message.

Type: String

SenderAddress

Description: This read/write property gets the address of the sender of the message.

Type: String

Read

Description: This read/write property gets or sets whether or not this message has been opened.

Type: Boolean

Methods

GetHTMLBody

Description: Returns the HTML part of the message in a string.

Syntax: GetHTMLBody()

Parameters: None

Return type: String.

GetTextBody

Description: Returns the Text part of the message in a string.

Syntax: GetTextBody()

Parameters: None

Return type: String.

GetTextBodyAsHTML

Description: Returns the Text part of the message in a string. The string is HTML encoded.

Syntax: GetTextBodyAsHTML

Parameters: None

Return type: String.

MoveToFolder

Description: Moves this message to the passed in folder.

Syntax: MoveToFolder(folder)

Parameters: AWMFolder object, which represents the folder to which the message should be moved.

Return type: Boolean. True if the message was successfully moved, False otherwise.

Delete

Description: Deletes this message

Syntax: Delete

Parameters: None.

Return type: Boolean. True if the message was successfully deleted. False otherwise.

GetToRecipients

Description: Returns a string, which lists the To: recipients of the message. If there is more than one recipient then each recipient is separated by a ‘;’ character.

Syntax: GetToRecipients()

Parameters: None.

Return type: String.

GetCCRecipients

Description: Returns a string, which lists the Cc: recipients of the message. If there is more than one recipient then each recipient is separated by a ‘;’ character.

Syntax: GetCCRecipients()

Parameters: None.

Return type: String.

GetFirstAttachment

Description: Returns a WMAttachment object, which is the first attachment of the message.

Syntax: GetFirstAttachment()

Parameters: None.

Return type: A WMAttachment object, or Nothing if the message does not have any attachments.

GetNextAttachment

Description: Returns a WMAttachment object, which is the next attachment of the message, after the passed in attachment.

Syntax: GetFirstAttachment(prevAttachment)

Parameters: prevAttachment is a WMAttachment, which is the attachment object that was returned from a previous call to GetFirstAttachment or GetNextAttachment.

Return type: A WMAttachment object, or Nothing if the message does not have any attachments.

WMMessageList

The WMMessageList object encapsulates a snapshot of all sent or received messages in WorkgroupMail.

More:

Properties

Methods

Properties

The WMMessageList object has no properties.

Methods

GetFirstMessage

Description: Gets the first message in the message list.

Syntax: GetFirstMessage()

Parameters: None

Return type: A WMMessage object which is the first message in the list.

GetNextMessage

Description: Gets the next message in the message list.

Syntax: GetNextMessage(message as object)

Parameters: message is a WMMessage object returned by a previous call to GetFirstMessage() or GetNextMessage()

Return type: A WMMessage object which is the next message in the list

GetLastMessage

Description: Gets the last message in the message list.

Syntax: GetLastMessage()

Parameters: None

Return type: A WMMessage object which is the last message in the list.

GetPreviousMessage

Description: Gets the previous message in the message list.

Syntax: GetPreviousMessage(message as object)

Parameters: message is a WMMessage object returned by a previous call to GetLastMessage() or GetPreviousMessage()

Return type: A WMMessage object which is the previous message in the list

GetCount

Description: Gets the number of messages in the message list.

Syntax: GetCount()

Parameters: None

Return type: Long number of messages

SortByDate

Description: Sorts the messages in the list by date.

Syntax: SortByDate(bAscending)

Parameters: bAscending - A boolean which should be set to true if the list is to be sorted in an ascending fashion or false if the list is to be sorted in a descending fashion.

Return type: Nothing

SortBySubject

Description: Sorts the messages in the list alphabetically by subject.

Syntax: SortBySubject(bAscending)

Parameters: bAscending - A boolean which should be set to true if the list is to be sorted in an ascending fashion or false if the list is to be sorted in a descending fashion.

Return type: Nothing

SortBySender

Description: Sorts the messages in the list alphabetically by sender name.

Syntax: SortBySender(bAscending)

Parameters: bAscending - A boolean which should be set to true if the list is to be sorted in an ascending fashion or false if the list is to be sorted in a descending fashion.

Return type: Nothing

SortByRecipients

Description: Sorts the messages in the list alphabetically by recipient name.

Syntax: SortByRecipients(bAscending)

Parameters: bAscending - A boolean which should be set to true if the list is to be sorted in an ascending fashion or false if the list is to be sorted in a descending fashion.

Return type: Nothing

WMComposeMessage

The WMComposeMessage object encapsulates a message being sent by the WorkgroupMail API.

More:

Properties

Methods

Properties

The WMComposeMessage object has no properties.

Methods

SetSubject

Description: Sets the subject of the message being composed.

Syntax: SetSubject(string sSubject)

Parameters: sSubject is the new subject string.

Return type: Boolean indicating success or failure.

SetMessageText

Description: Sets the message body text of the message being composed.

Syntax: SetMessageText(string sMessageText)

Parameters: sMessageText is the new message string.

Return type: Boolean indicating success or failure.

SetSender

Description: Sets the sender of the message being composed.

Syntax: SetSender(string sName, string sAddress)

Parameters: sName is the name of the sender.

sAddress is the sender's address.

Return type: Boolean indicating success or failure.

AddRecipient

Description: Adds a recipient to the new message. The message will be sent to all added recipients.

Syntax: AddRecipient(string sName, string sAddress)

Parameters: sName is the name of the recipient.

sAddress is the recipient's address.

Return type: Boolean indicating success or failure.

AddRecipients

Description: Adds one or more recipients to the new message. If there is more than one recipient, each recipient must be separated by a ‘;’ character.

Syntax: AddRecipient(string sRecipients, integer iType)

Example: message.AddRecipients(“fred@bloggs.com;John <john@domain.com>”)

Parameters: sRecipients is the ‘;’ separated list of recipient names and addresses.

iType specifies whether the recipients should be added to

the To:, Cc or Bcc list. The values are: 0 = To, 1 = Cc, 2 = Bcc.

Return type: Boolean indicating success or failure.

AddAttachment

Description: Adds an attachment to the message being composed.

Syntax: AddAttachment(string sDisplayName, string sFilename)

Parameters: sDisplayName is the filename of the attachment without it's path that you wish the receiver of the message to see.

sFilename is the full path and filename of the file being attached.

Return type: Boolean indicating success or failure.

Send

Description: Send the composed message. The message must have at least one recipient and a sender address.

Syntax: Send()

Parameters: None

Return type: Boolean indicating success or failure.

Send

Description: Send the composed message. The message must have at least one recipient and a sender address.

Syntax: Send()

Parameters: None

Return type: Boolean indicating success or failure.

SendKeepCopy

Description: Sends the composed message, keeping a copy of the sent message in the user's Sent Items personal folder. The message must have at least one recipient and a sender address.

Syntax: SendKeepCopy()

Parameters: None

Return type: Boolean indicating success or failure.

SendUsingSMTP

Description: Sends the composed message by establishing a connection to the passed in server on the passed in port number (the default port number should be 25) and sending via SMTP. The message must have at least one recipient and a sender address.

Syntax: SendUsingSMTP(sServer, iPort)

Parameters: sServer - A string which represents the IP address or server name of the SMTP server.

iPort - The port number on which an SMTP session will be established. This should nearly always be set to 25.

Return type: Boolean indicating success or failure.

SendUsingSMTPKeepCopy

Description: Sends the composed message by establishing a connection with the passed in server on the passed in port number (the default port number should be 25) and sending via SMTP, keeping a copy of the sent message in the user's Sent Items personal folder. The message must have at least one recipient and a sender address.

Syntax: SendUsingSMTPKeepCopy(sServer, iPort)

Parameters: sServer - A string which represents the IP address or server name of the SMTP server.

iPort - The port number on which an SMTP session will be established. This should nearly always be set to 25.

Return type: Boolean indicating success or failure.

SendMIMEFileUsingSMTP

Description: Sends the passed in MIME file by establishing a connection with the passed in server on the passed in port number (the default port number should be 25) and sending via SMTP.

Syntax: SendMIMEFileUsingSMTP(sMIMEFile, bKeepCopy, sServer, iPort)

Parameters: sMIMEFile - A string which represents the filename of a file which contains a MIME formatted message.

bKeepCopy - a boolean that indicates whether or not a copy of the sent message should be kept in the user's Sent Items folder.

sServer - A string which represents the IP address or server name of the SMTP server.

iPort - The port number on which an SMTP session will be established. This should nearly always be set to 25.

Return type: Boolean indicating success or failure.

SendMIMEFile

Description: Sends the passed in MIME file.

Syntax: SendMIMEFile(sMIMEFile, bKeepCopy)

Parameters: sMIMEFile - A string which represents the filename of a file which contains a MIME formatted message.

bKeepCopy - a boolean that indicates whether or not a copy of the sent message should be kept in the user's Sent Items folder.

Return type: Boolean indicating success or failure.

WMAttachment

The WMAttachment object encapsulates a message attachment.

More:

Properties

Methods

Properties

This object does not contain any properties.

Methods

GetFileName()

Description: Gets the display name for the attachment.

Syntax: GetFileName()

Return type: String

GetFileSize()

Description: Returns the size of the attachment in bytes.

Syntax: GetFileSize()

Return type: Integer

GetFileSizeString()

Description: Returns a string which represents the size of the attachment. The units of the file size will change according to the size, i.e. 2Mb, rather than 2048 Kb, or 1Kb, rather than 1024 bytes.

Syntax: GetFileSizeString()

Return type: String

Decode()

Description: Decodes the attachment to a specified folder.

Syntax: Decode()

Return type: String, which points to a filename of the actual decoded attachment file.

GetEmbedded()

Description: Returns True if this attachment is an embedded reference inside the message rather than a separate file attachment.

Syntax: GetEmbedded()

Return type: Boolean.

WMSpamServer

The WMSpamServer object encapsulates a WorkgroupMail local user and provides properties and methods to interrogate and edit user details. Various methods and properties of the WMSession object and other objects return a WMSpamServer object and take WMSpamServer objects as arguments.

More:

Properties

Methods

Properties

Name

Description: The name given to this spam server.

Type: String

ServerAddress

Description: The server address for this spam server.

Type: String

Enabled

Description: True if this spam server is enabled. False otherwise.

Type: Boolean

Methods

GetID()

Description:	Gets the unique identifier of this spam server.
Syntax:	GetID ()
Return type:	Integer

Glossary of Terms

Plug-in

Interface

Virtual mailbox

ISP

Plug-in

A DLL file that may be "plugged in" to WorkgroupMail in order to provide additional processing on each incoming and outgoing message.

Interface

In the context of WorkgroupMail, an Interface is the IP address of a network card.

Virtual mailbox

A mailbox that can receive mail sent to a certain e-mail address, for example, sales@company.com . Messages sent to a virtual mailbox must always be forwarded to a WorkgroupMail user or group. Virtual mailboxes may be configured to auto-respond to messages sent to it.

ISP

Internet Service Provider. An organization, such as AOL, that hosts an organization's e-mail.