



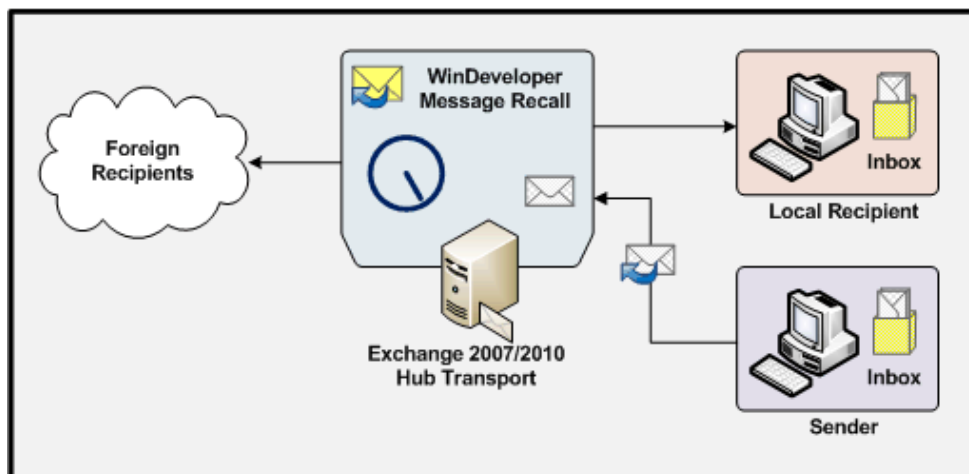
time to make things
right

WinDeveloper Message Recall v2.0

Contents

| | |
|---|----|
| 1. Message Recalling Works! Here is how..... | 3 |
| 2. WinDeveloper vs Native Exchange Message Recalling..... | 4 |
| 3. System Setup..... | 6 |
| 3.1 Minimum Requirements | 6 |
| 3.2 MS Exchange Server Roles | 6 |
| 3.3 Fresh Installation..... | 7 |
| 3.4 Upgrading from WinDeveloper Message Recall v1.x..... | 9 |
| 4. Recall Processing Configuration..... | 10 |
| 4.1 Main Recall Processing Options | 11 |
| 4.1.1 Retain emails queued for (seconds) | 12 |
| 4.1.2 If unable to Recall email..... | 12 |
| 4.1.3 Configuration Overrides..... | 13 |
| 4.1.4 Creating/Editing Overrides | 13 |
| 4.1.5 Configuration Examples | 15 |
| 4.2 Archiving | 21 |
| 4.3 Logging | 22 |
| 4.3.1 Log File Fields | 23 |
| 4.4 Status Reporting..... | 24 |
| 4.5 Enable WinDeveloper Message Recall for Individual Users | 25 |
| 5. Working with WinDeveloper OWA Recall..... | 27 |
| 5.1 OWA Premium | 27 |
| 5.2 The OWA Recall Interface | 28 |
| 6. Licensing WinDeveloper Message Recall..... | 30 |
| 7. Contacting WinDeveloper | 31 |

1. Message Recalling Works! Here is how...



WinDeveloper Message Recall dramatically increases your ability to successfully recall emails. Most often you realize of a mistake as soon as you hit the Send button. This is when WinDeveloper Message Recall provides you with a golden opportunity to make things right.

WinDeveloper Message Recall is a server side solution, requiring no new client software. Outlook users [submit recall requests using the standard Recall This Message functionality](#). WinDeveloper also provides an extension for OWA users, the [WinDeveloper OWA Recall extension](#). This adds the Recall This Message functionality to Outlook Web Access/App, allowing users to submit recall requests straight from their browser.

The WinDeveloper Message Recall engine must be installed on each Exchange 2007/2010 Hub Transport server. This is where recall requests are processed and fulfilled.

When a local user sends an email, WinDeveloper Message Recall holds the email within the transport queue for a preset number of seconds. During this time the user is able to recall the email and WinDeveloper will be able to satisfy the request for all recipients (even foreign recipients). Otherwise if the queue hold time elapses the email is released for final delivery.

From our research, even if emails are held at the queue for as little as 30 seconds you already manage to satisfy a significant number of recall requests. In practice users will very often realize a mistake was committed as soon as they click Send. At this point opening the email from Sent Emails and clicking "Recall Message" only takes a few seconds.

Recalling emails, by its very own nature is a race against time. There is no way to change that fact. In case a user does not immediately realize an error was committed, chances are that the end recipient opens the email. Unfortunately no software solution can change this fact. So at WinDeveloper we focused on the time window when recalling emails makes sense and have provided a solution that is predictable and can be relied on.

2. WinDeveloper vs Native Exchange Message Recalling

How many times have you sent an email too early, missing an attachment, or even worst to the wrong recipient? You hit send and immediately realize the mistake. Sure, you can try recalling it from Outlook. But will that work?

The message recall functionality available in Exchange hardly ever works. Indeed recall requests very often have the opposite effect. It is like telling recipients to open the email and spot the error.

WinDeveloper Message Recall gives you a predictable process for recalling sent emails. This is superior to the MS Exchange functionality in many ways. Here is why:

- **Process Recall Messages at the Server** – This is a lot more efficient than the MS Exchange implementation where recalls are processed at the recipient Outlook client.
- **Recipients Never See the Email** – If successful, neither the original email nor the recall request ever reaches the email recipients. Thus the recipient will never know. Compare that to the native recall system where Outlook may be configured not to automatically process recalls.
- **Works with both Local and Foreign Recipients** – Recalling emails addressed to recipients outside the organization is often more critical than for local recipients. This is when the native Exchange solution let's you down completely. Not so with WinDeveloper, all emails are handled equally well.
- **Recall Emails Directly from OWA** – MS Exchange does not provide the ability to recall emails from the Outlook Web Access/App browser interface. WinDeveloper fills this gap with the included [WinDeveloper OWA Recall extension](#).
- **Accurate Success/Failure Notifications** – WinDeveloper generates notifications identical to those of the native Exchange solution. What's better is that on succeeding, email delivery is blocked for all recipients. This is not so with native recalling, here a recall may succeed for some of the recipients but fail for others.
- **No New Client-Side Software** – No new software to install and manage on client computers. Users [recall emails directly from MS Outlook](#) and the [OWA web interface](#).
- **Log all Recall Attempts** – Centralized record of all recall requests. Logs include all the key information such as the subject, sender, recipient list, and whether or not the recall was successful.
- **Archive Emails Deleted by Recalls** – Email senders might want to block recipients from seeing their email. However if the organization policy dictates that deleted emails are to be archived at the server, WinDeveloper will also satisfy that requirement.

- **Different Recall Settings for Different User Groups** – Recalling emails addressed to recipients outside the organization is often more critical than for local recipients. To reflect this fact different Sender/Recipient groups may be assigned different recall handling settings.
- **Drop Failed Recall Requests or Fallback to Traditional Recalling** – If unable to satisfy a recall request you may choose to delete the request at the server. Thus recipients remain unaware of the recall attempt. Alternatively you may choose to deliver the recall request to the recipients, falling back to traditional recalling.

3. System Setup

WinDeveloper Message Recall is a server side application. The application has to be installed on each Exchange 2007/2010 Hub Transport server. In this manner it is able to process all emails whether these are addressed to local or foreign recipients.

3.1 Minimum Requirements

1. **Platform Support** - WinDeveloper Message Recall is a 64-bit application and supports all x64 Windows Server platforms. This includes Windows 2003, Windows 2008 and Windows 2008 R2. X64 Small Business Server SBS packages are also supported.
2. **Microsoft Exchange Support** - WinDeveloper Message Recall supports Exchange server versions 2007 and 2010.

3.2 MS Exchange Server Roles

WinDeveloper Message Recall is made up of two components:

- **WinDeveloper Message Recall Engine** – Responsible for processing recall requests. Must be installed on each Exchange 2007/2010 Hub Transport Server.
- **WinDeveloper OWA Recall extension** – Adds the functionality to recall messages directly from the Outlook Web Access/App interface. Must be installed on each Exchange 2007/2010 Client Access Server.

The MS Exchange 2007/2010 functionality is split into server roles that include the Hub Transport and Client Access server amongst others. Organizations running Exchange on a single server, simply install all WinDeveloper Message Recall components on this server.

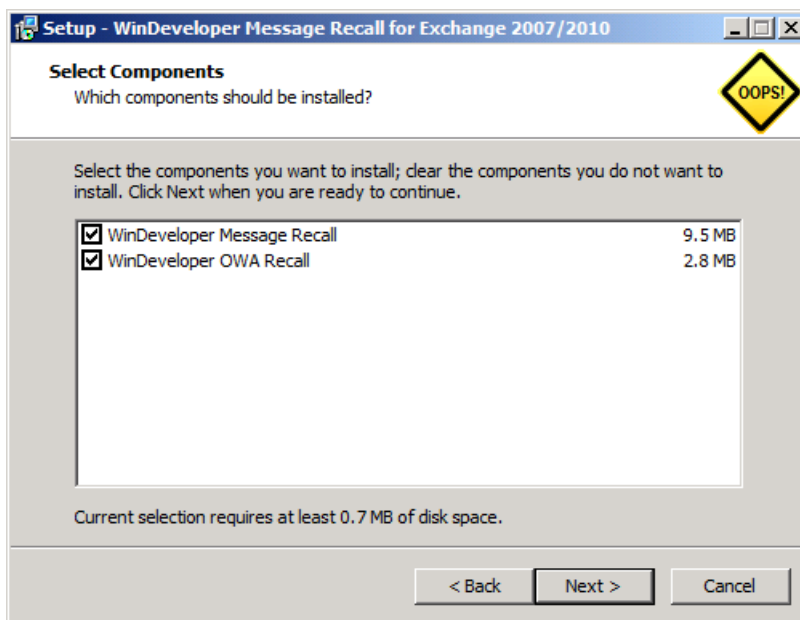
Organizations running MS Exchange roles distributed over multiple servers should run the WinDeveloper Message Recall installation on each server running at least one of the Client Access or Hub transport roles. Wherever you have a Hub Transport the Recall Engine has to be installed. Likewise OWA Recall must be installed on each Client Access server.

Note: *OWA Recall requests can only be fulfilled by the WinDeveloper Message Recall engine. If OWA Recall is installed without the Recall Engine, any recall requests are delivered to the recipients as regular emails.*

3.3 Fresh Installation

The WinDeveloper Message Recall setup wizard is very easy to follow. Here we take a quick look at the key wizard steps.

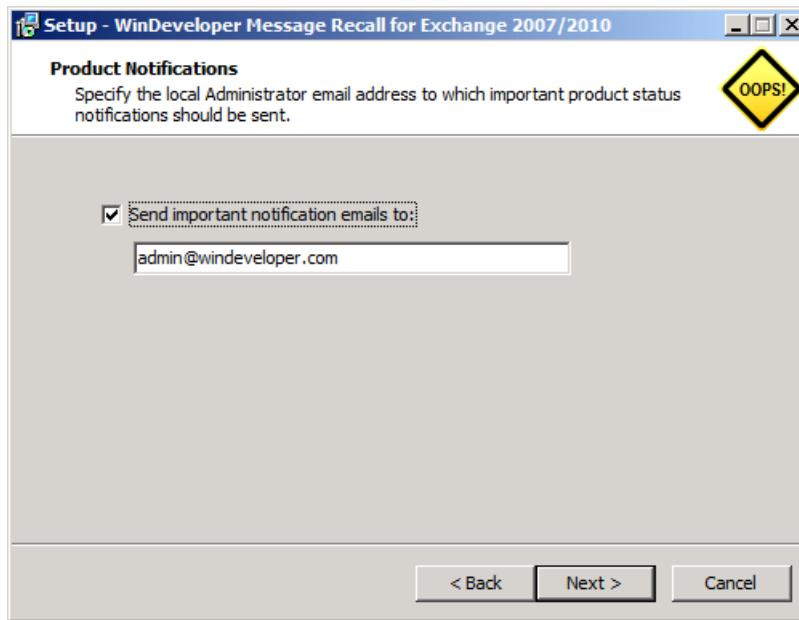
Starting the installation, and moving forward through the initial steps, the wizard presents the Component Selection page:



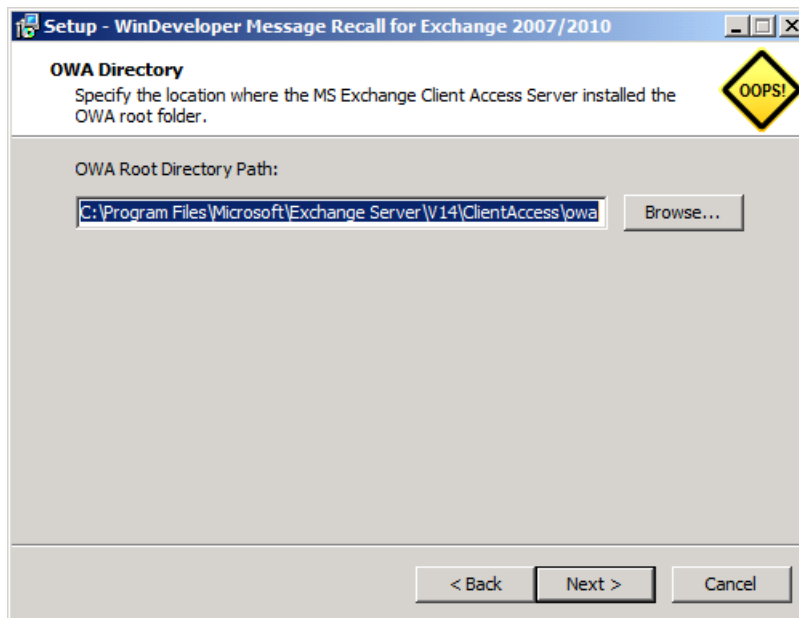
At this step select the components to install depending on the Exchange server roles running on the machine. **WinDeveloper Message Recall** is the Recall Engine and is installed on Hub Transport servers. **WinDeveloper OWA Recall** is the OWA extension and is installed on Client Access servers.

Next click your way through the wizard steps that follow.

If the Engine component was select, the Wizard will prompt you for an email address to which notifications are to be sent. These alert the administrator of important events such as the expiry of an evaluation license.



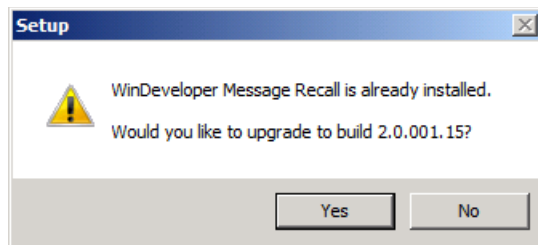
If the OWA Recall extension component was selected, the Wizard will also prompt for the full path to the OWA root directory. The Wizard will automatically detect this path itself and show it for confirmation purposes:



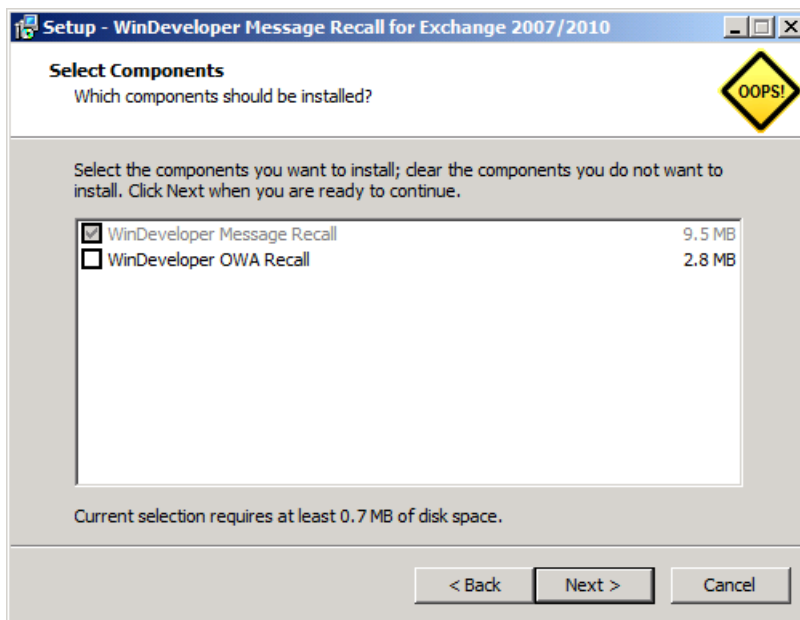
These are the key installation steps. Completing these will allow you to successfully install WinDeveloper Message Recall.

3.4 Upgrading from WinDeveloper Message Recall v1.x

To upgrade to the latest version you can run WinDeveloper Message Recall v2.0 directly on the machines where v1.x is currently installed. Setup detects v1.x on startup and prompts if it should proceed with an upgrade.



The Recall Engine component is selected automatically for upgrading. The wizard also provides the option to add the OWA Recall extension. This should be added in case both Hub and Client Access servers are running on the same machine.



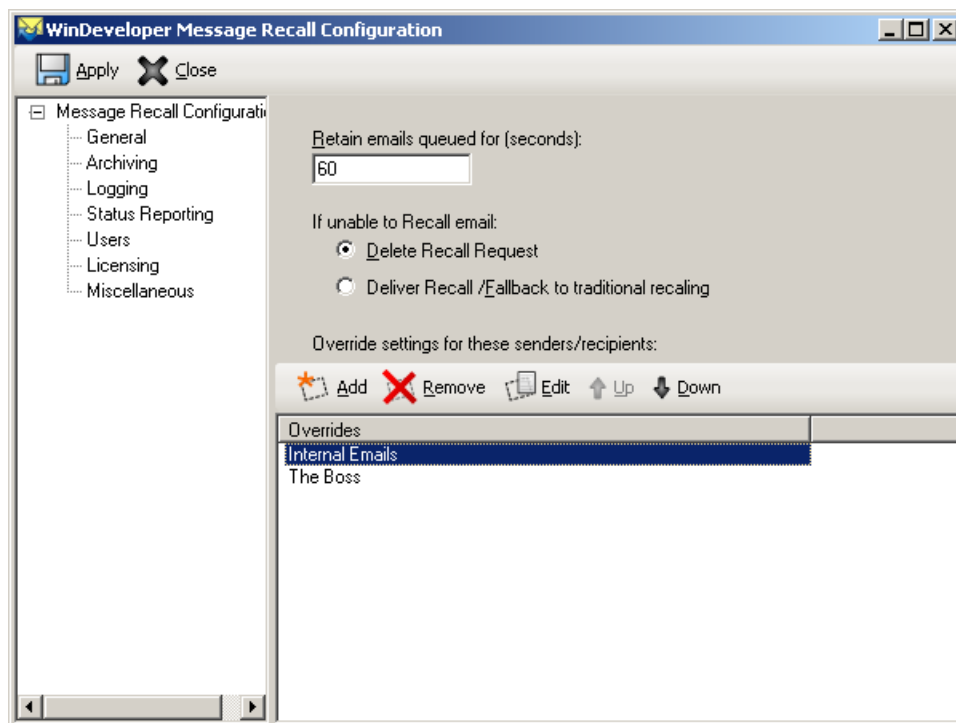
From there follow the wizard to complete the upgrade process for each v1.x installation.

In case of more Exchange servers running the Client Access or Hub Transport roles make sure to also run the installation on these machines as well.

4. Recall Processing Configuration

The Configuration interface is available from the application program group under:
Windows Start | Programs | WinDeveloper Message Recall | Message Recall

The configuration is organized in two panes. The left pane shows the main configuration categories while the right pane shows the options for the currently selected category.



General - provides the main set of options that determine how messages are processed

Archiving - allows us to retain a copy on disk of all successfully recalled emails

Logging - enables detailed CSV logging of all processed recall requests

Status Reporting - enables alerting the administrator of important error conditions that require administrative attention

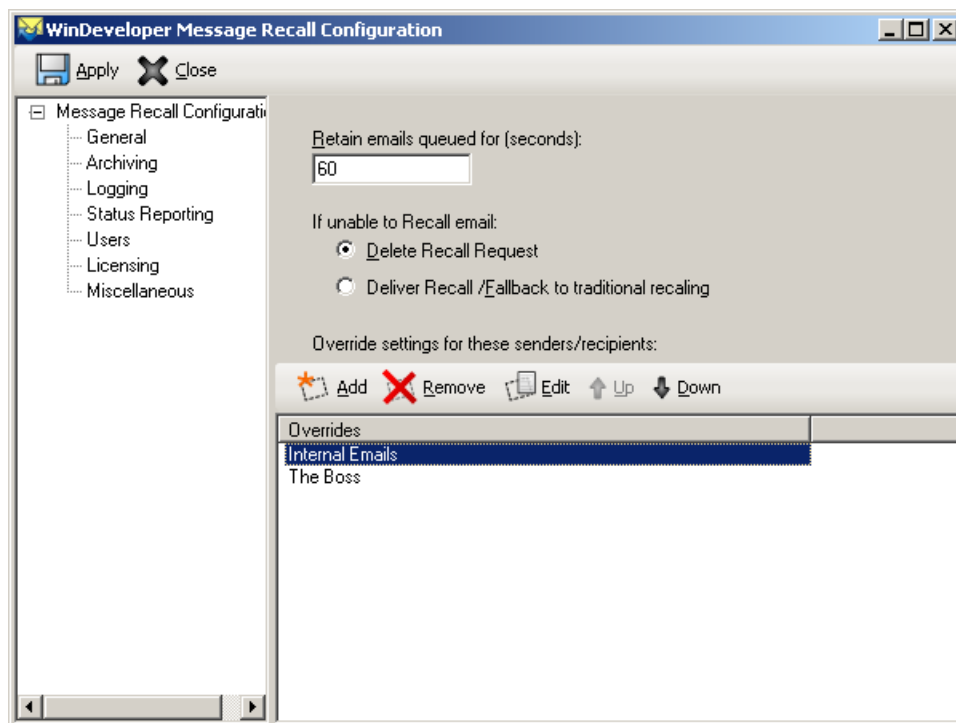
Users - enables/disables the product for individual users and SMTP domains

Licensing - shows the type of license currently in use such as evaluation, single server license, etc.

Miscellaneous - gives quick access to product version and contact information

4.1 Main Recall Processing Options

The **General** category provides the main set of options that determine how messages are processed.



When configuring the product it is good to have a basic idea of how the product works as explained in [1. Message Recalling Works! Here is how...](#)

Here we focus on the key scenario the product is designed to address, the case when a user sends an email and immediately realizes that an error was committed. This is by far the most common case when message recalling is required. Arguably this is often considered the only case when we can realistically expect to cleanly recall an email.

For this to work, WinDeveloper Message Recall will hold newly submitted emails within the transport queue for a preset number of seconds. If no recall request is submitted during this time the email is released for final delivery. Thereafter WinDeveloper Message Recall will no longer be able to satisfy any request for this message.

From this discussion it should be clear that the most important parameter is the length of time emails are held at the queue. On one hand we need to give enough time for users to submit the recall request, on the other we want to avoid holding emails unnecessarily.

Note: Only emails originating from local mailboxes are processed. Internet originating emails are not held at the queue.

Note: While emails are queued at the transport, no SMTP connections are wasted. Holding emails at the queue in this manner does not cause any significant resource loss.

4.1.1 Retain emails queued for (seconds)

Retain emails queued for (seconds) specifies the length of time emails are to be held at the transport queue for recalling purposes.

This is the default value to be applied to all emails sent from local users. This default may be overridden for specific email categories. For example internal emails may be assigned a different time value from that applied to emails addressed to foreign recipients. Check the Overrides configuration for more details on this.

Our research showed that users, who already know how to recall an email, normally require less than 30 seconds to submit a recall request. The operation involves opening the email from Sent Items and clicking on 'Recall This Message'. Even so, we should keep in mind the anxiety a user goes through when facing the risk of information disclosure. This may cause him to act slower than average. Based on these considerations we set the initial queue retention length to 60 seconds, giving ample time for an email to be recalled.

4.1.2 If unable to Recall email

If WinDeveloper Message Recall manages to successfully block the delivery of a recalled email, both the original and recall request emails are of course deleted.

A question arises on what should be done if the recall cannot be satisfied. This is what the options under **If unable to Recall email** determine:

Delete Recall Request – The request is deleted at the server. In this manner recipients will never know that an attempt was made for the email to be recalled. It is a known fact that recall requests in Exchange tend to have the opposite effect. As soon as a recipient sees such a request, he quickly opens the original email to spot the error. So aborting the delivery of these requests is intended to avoid attracting too much attention.

Deliver Recall/Fallback to traditional recalling – The alternative is that of letting the recall request through. Here we go back to traditional recalling where recipients receive an email request telling them that the sender would like to recall the message. With this option we are combining the two recalling solutions together.

4.1.3 Configuration Overrides

Sometimes it is desirable to apply different settings to different email categories. Here are some points to consider:

1. An email sent in error to a foreign recipient is potentially more damaging than an email addressed to an internal recipient. Thus emails addressed to foreign recipients may benefit from a longer recall time window.
2. When internal emails are involved users may expect quicker delivery. Errors may be more tolerated as long as these stay within the organization boundaries.
3. Some users because of their work will handle more confidential information than others. These may also benefit from a longer recall time window.

The Override list is specifically intended to handle the different requirements of these email categories. Each Override provides unique settings for the main processing parameters discussed in:

- [Retain emails queued for \(seconds\)](#)
- [If unable to Recall email](#)

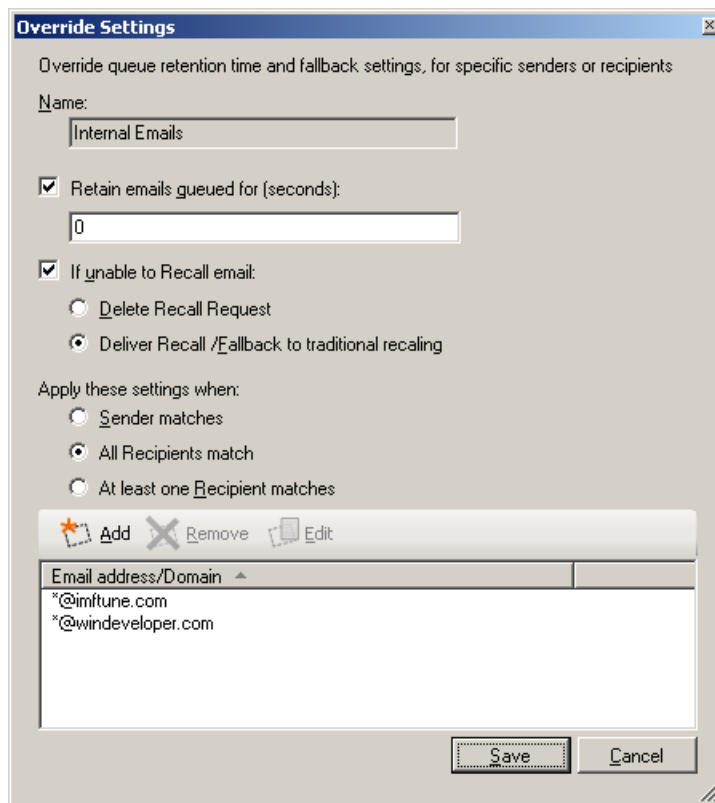
Each Override includes a list of sender/recipient email addresses. These are used to identify to which emails the Override settings are to be applied.

If multiple Overrides are configured, WinDeveloper Message Recall goes through these in the order shown at the configuration, from top to bottom. The Override address list is tested against the email being processed. If matched the Override settings are applied, otherwise the next Override is considered until the list is exhausted. If none of the Overrides match, the main processing parameters configured under the General category are applied.

4.1.4 Creating/Editing Overrides

Working with the Overrides list is very simple. On top of the list we have the Add, Remove, Edit, Up and Down buttons. The first three buttons allow us to create, delete and change the Override settings. The Up and Down buttons allow us to change the Override processing order. It is possible for an email to match more than one Override. In that case the top most Override is applied.

Clicking on the Add button opens the Override Settings dialog shown below.



This dialog reproduces the settings shown directly under the General category:

- [Retain emails queued for \(seconds\)](#)
- [If unable to Recall email](#)

Here these settings are meant to replace the settings configured under the General category whenever the email Sender/Recipients matches the addresses listed at the bottom.

For us to override **Retain email queued for (seconds)**, we simply select the checkbox next to the field and specify a new value. Same goes for **If unable to Recall email**.

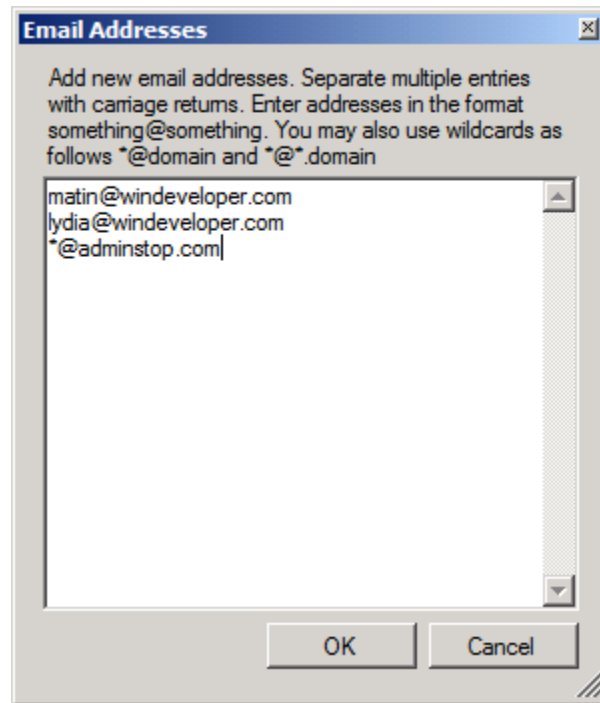
The bottom most set of options identify the email category to which this Override applies. We can choose to identify emails by sender or by recipient. To do that, select one of:

Sender matches – The Override is applied when the email sender matches the address list.

All Recipients match – The Override is applied when all email recipients match the address list.

At least one Recipient matches – The Override is applied when one or more recipients match the address list.

Once we select how we want to identify the Override email category, click Add to enter the list of addresses/domains to be matched.



Likewise click Delete or Edit to update the address list as necessary.

4.1.5 Configuration Examples

To better appreciate how to configure WinDeveloper Message Recall it is best to have a look at some examples.

4.1.5.1 Example 1 - Different Recall Time Windows for Internal and Foreign Emails

Two very important email categories that often require different Message Recall handling are:

1. **Internal Emails** – Emails sent from one local user to other local users.
2. **Foreign Emails** – Emails sent from one local user to foreign recipients outside the organization.

Let's say we want to allow for a recall time window of 2 minutes in case of foreign emails. However we don't want to hold internal emails for that long, so we shorten the time window to 30 seconds when internal emails are involved.

To do this we need a method to clearly identify Internal Emails. Here we use the organization SMTP domains. These are the domains used to assign emails addresses to our user mailboxes. In this example our domains are:

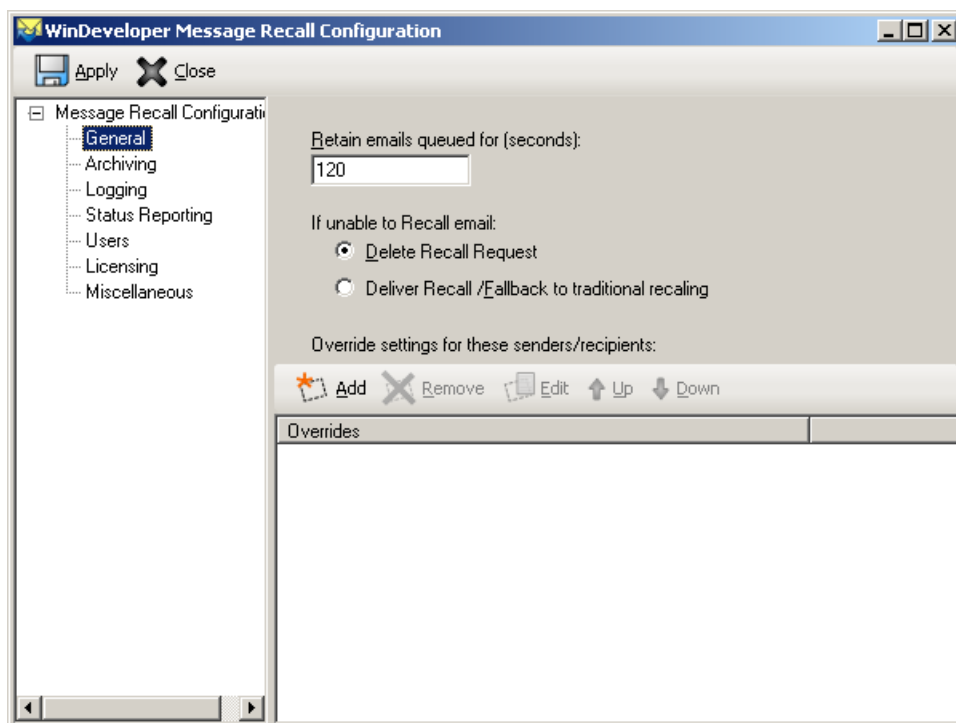
windeveloper.com

imftune.com

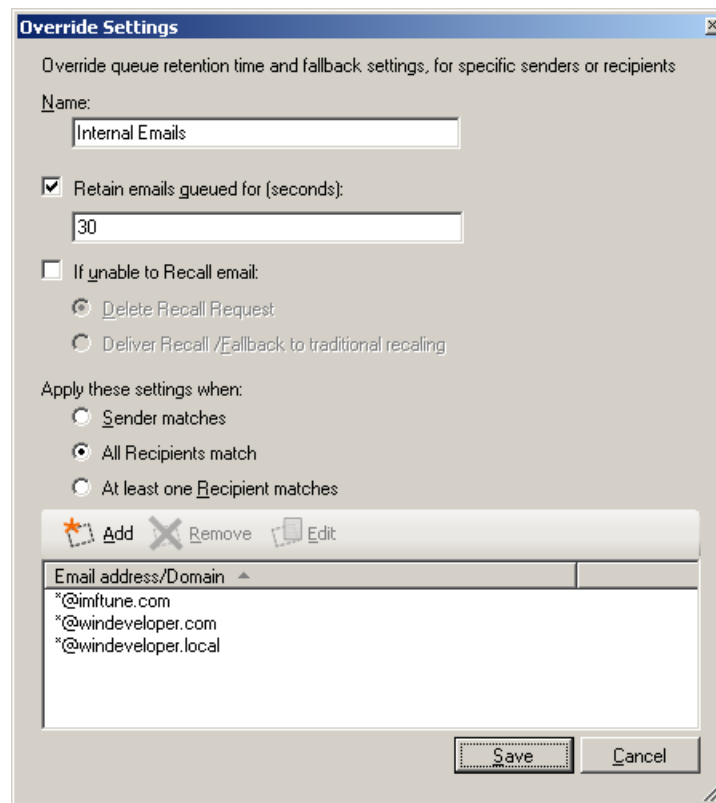
windeveloper.local

Configuration Steps:

1. Open the Configuration and select the General category
2. Under ***Retain emails queued for (seconds)***, set the time windows length to 2 minutes so we enter the value: **120**. Note that this is the time window Foreign Emails require. So next we will need to create an Override for Internal Emails not to be held that long.



3. Click Add at the Override list and configure this as follows:
 - Name: ***Internal Emails***
 - Select Checkbox for: ***Retain emails queued for (seconds)***
 - Set the time window for internal emails to **30** (seconds).
 - Select ***Apply these settings when - All Recipients match***
 - At the address list enter the 3 local domains as follows:
 - *@windeveloper.com**
 - *@imftune.com**
 - *@windeveloper.local**



4. Save Changes

With these settings WinDeveloper Message Recall will apply the Override whenever all recipient email addresses match a local domain. If at least one recipient does not match, we are dealing with a Foreign Email and the global settings configured at the main Configuration category are applied.

Note: When configuring the product always keep in mind that Message Recall processing only applies to emails sent from local mailboxes. Emails received from outside are never processed, not even if their addresses match an Override.

Note: If we wanted *Internal Emails* to be immediately delivered without being held at the queue, we would set *Retain emails queued for (seconds)* to zero.

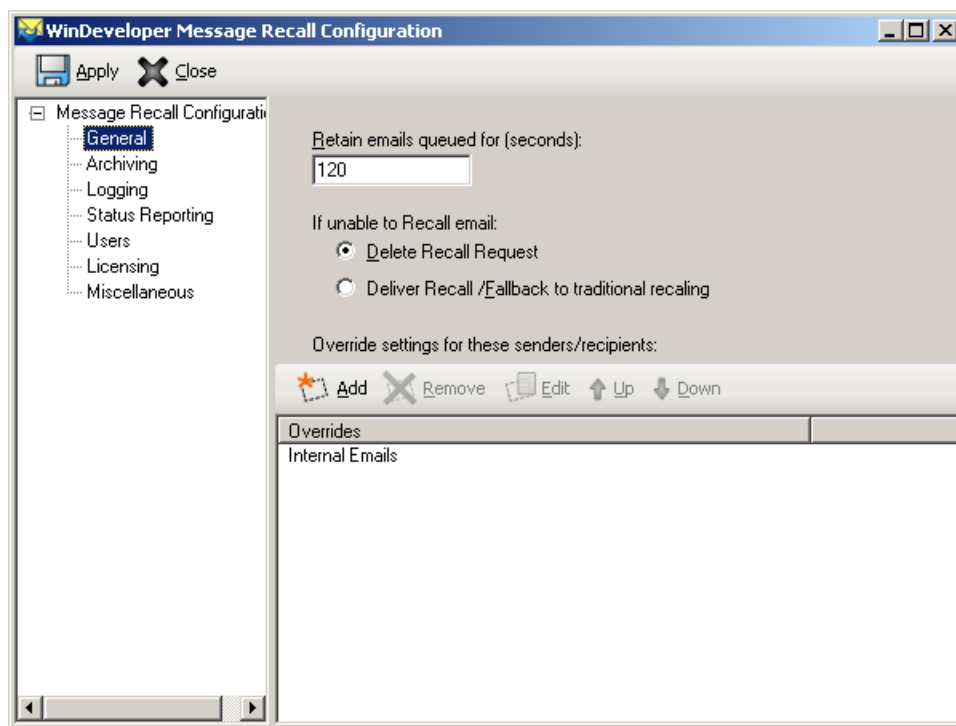
4.1.5.2 Example 2 – Extra Recalling Time for Specific Users

In this example we will continue to build on Example 1. Our initial configuration already includes the Override created for Internal Emails.

Let's consider the case when some users are handling confidential information all the time. These users might not afford an error, not even if it only involves internal recipients. In this manner we are defining a new email category. So this time we will need to cater for all of these:

1. **Special Users** – Users handling highly confidential information that will always require a 2 minute recall time window even in case of internal emails.
2. **Internal Emails** – All emails sent from one local user to another except when the sender is a Special User. These will have a 30 second recall time window.
3. **Foreign Emails** – Emails sent from one local user to a foreign recipient outside the Organization. These will have a 2 minute recall time window.

In example 1, we already catered for the Internal and Foreign Categories.



We now need to add another Override to cater for Special Users:

1. At the Configuration | General category click Add
2. Configure the Override as follows:
 - Name: **Special Users**
 - Select Checkbox for: **Retain emails queued for (seconds)**
 - Set the time window for Special Users. In this example we use **120** (seconds).
 - Under **Apply these settings when** select **Sender matches**
 - Lastly at the address list enter the email address for the users, for example:
HR@windeveloper.com

Override Settings

Override queue retention time and fallback settings, for specific senders or recipients

Name:

☒ Retain emails queued for (seconds):

☐ If unable to Recall email:

- ☒ Delete Recall Request
- ☐ Deliver Recall / Fallback to traditional recalling

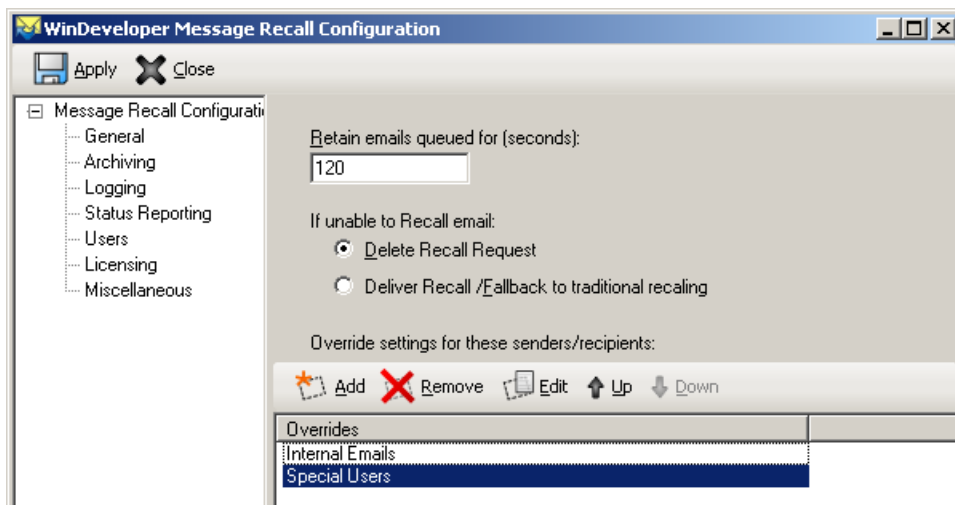
Apply these settings when:

- ☒ Sender matches
- ☐ All Recipients match
- ☐ At least one Recipient matches

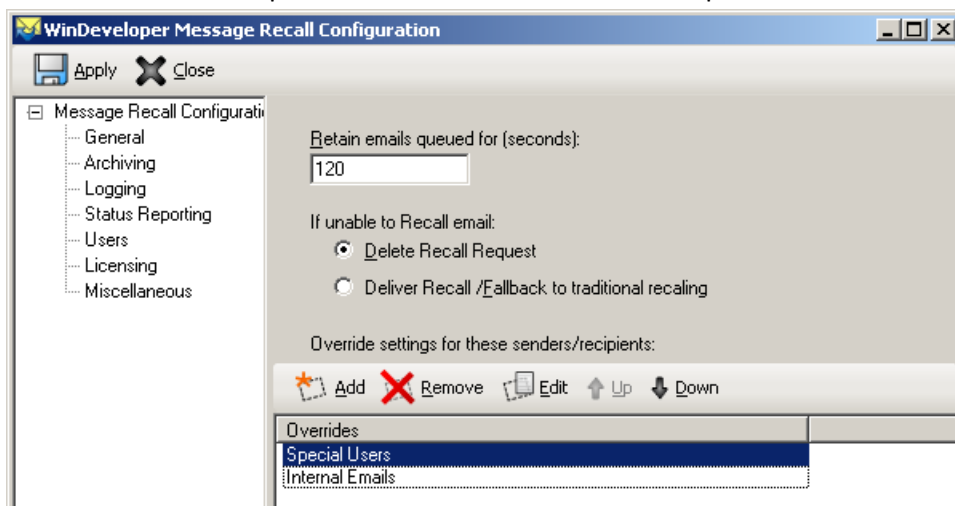
Email address/Domain

HR@windeveloper.com

3. Save the Override and go back to the General category.



4. From here select the Special Users Override and click the Up button so as to move it to the top.



5. Save Changes

The most interesting point in this example is the Override order. WinDeveloper Message Recall goes through Overrides from top to bottom and stops as soon as it finds a match.

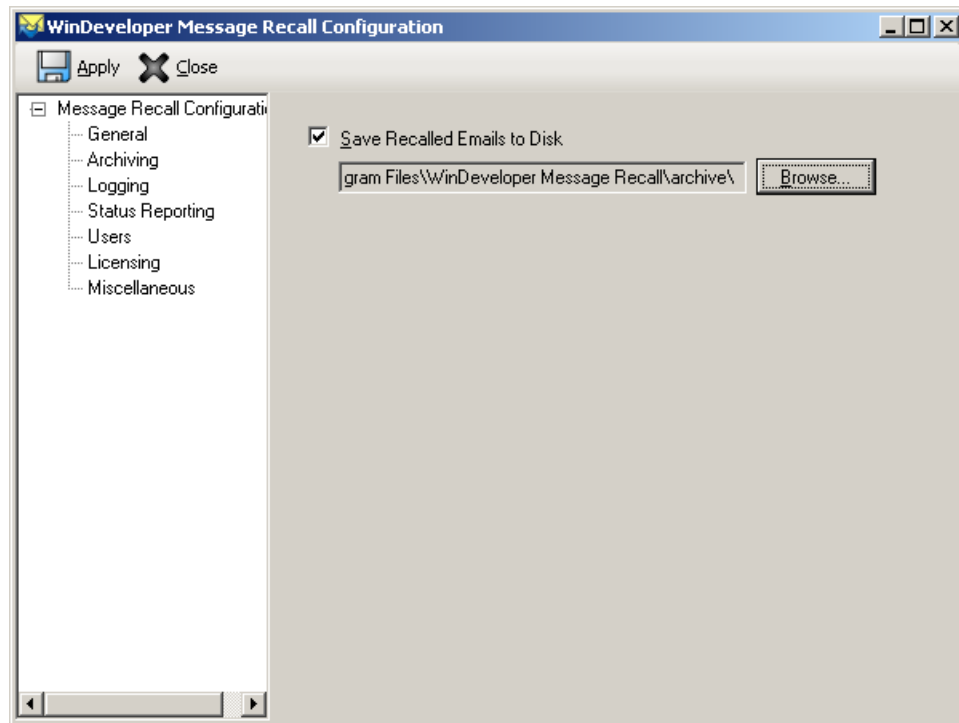
If we look closely at the **Special Users** and **Internal Emails** Overrides we see that the two overlap. Whenever a **Special User** sends an email internally both Overrides could be applied. Thus the order here is important as it determines which of the two is actually applied.

Special Users will always have a recall time window of 2 minutes, whether the email is addressed to internal or foreign recipients. So we put this Override on top to be considered first. If we don't do that, the **Internal Emails** Override would match internal emails before the **Special Users** Override is ever considered.

4.2 Archiving

Disk Archiving saves a copy of recalled emails to disk. Email senders might want to block recipients from seeing their email. However if the organization policy dictates that deleted emails are to be archived at the server, WinDeveloper will also satisfy that requirement.

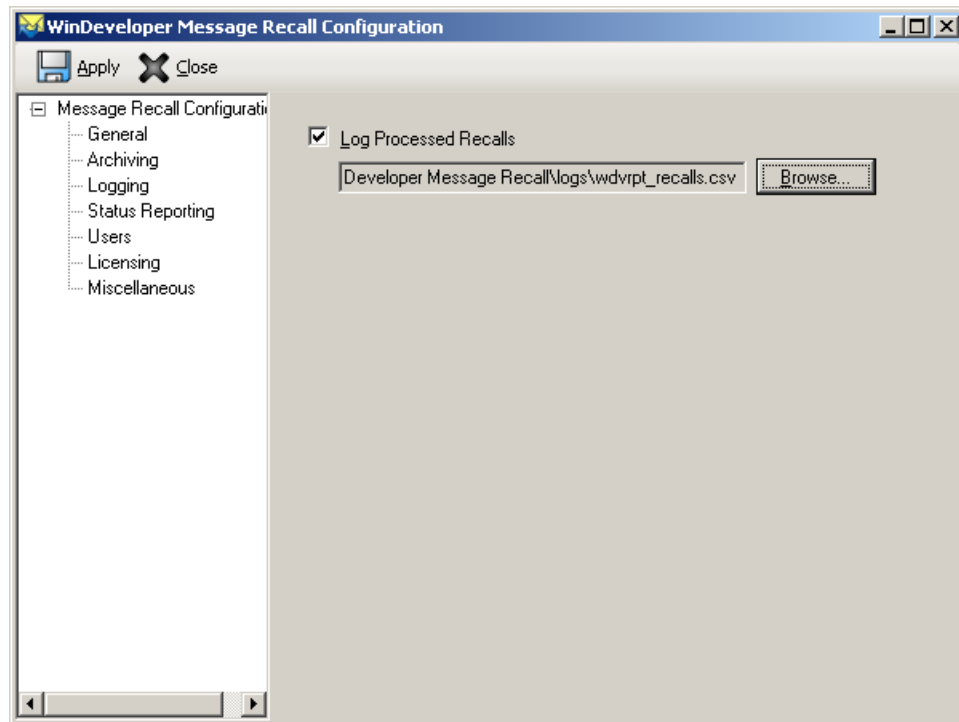
If necessary archived emails may be manually moved to the Exchange 2007/2010 Hub Transport Pickup directory for these to be delivered to their originally intended destination.



Here the configuration is very simple. The top checkbox enables/disables disk archiving. Following that we specify the archive directory path. For performance reasons this path should be located on a local disk. UNC paths and mapped drives are not supported.

4.3 Logging

Logging keeps record of processed recall requests in a CSV formatted log file. Logs include all the key information such as the subject, sender, recipient list, and whether or not the recall was successful.



Here the configuration is very simple. The top checkbox enables/disables logging. Following that we specify the log file path. For performance reasons this file should be saved to a local disk. UNC paths and mapped drives are not supported.

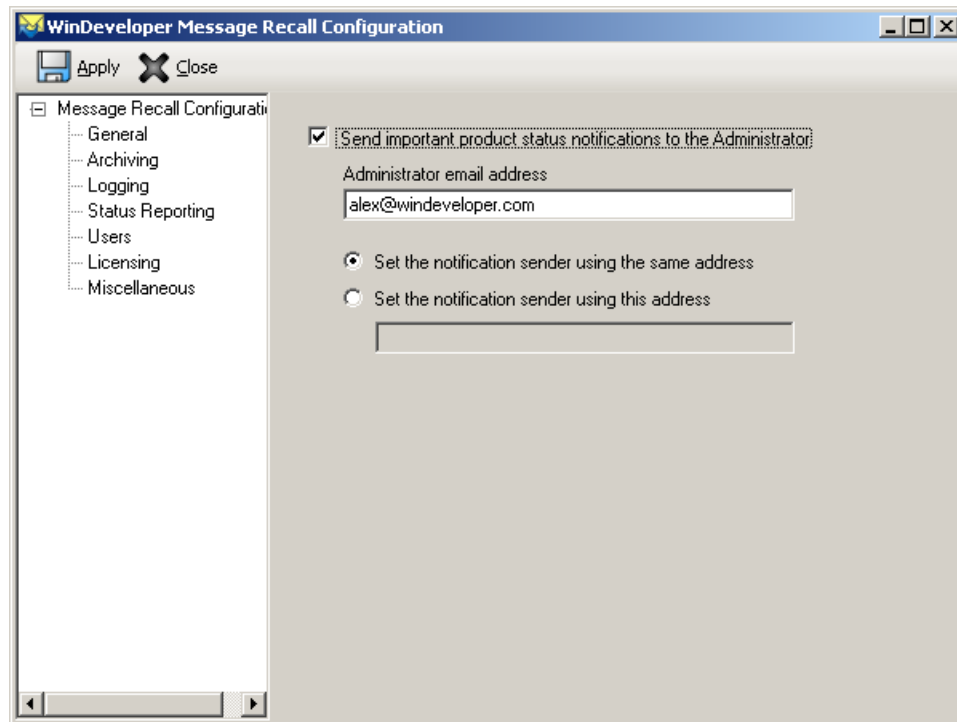
4.3.1 Log File Fields

The following table lists the fields logged by WinDeveloper Message Recall

| Field Name | Description |
|--------------|---|
| Date | Date when log entry was created |
| Time | Time when log entry was created |
| Result | Numeric value identifying the recall processing outcome: 0 – Email Recalled Successfully. 1 – Already Recalled Successfully. We submitted multiple recall requests. 2 – Failed to Recall. Most likely the request was sent too late. |
| Description | Textual description of the recall result. This takes one of the following values: <ul style="list-style-type: none"> • Success • AlreadyRecalled • Failed |
| Subject | Recall request email subject |
| Sender | Sender email address |
| Recipients | List of comma separated recipients |
| Transport | Describes how the recall request message itself was handled. This can be one of: <ul style="list-style-type: none"> • Delivered • Deleted <p>Whenever the original email is successfully recalled, the recall request is deleted and hence this field is set to Deleted. However if the recall request cannot be satisfied, the application checks the settings under: General If unable to recall email</p> <p>Delete Recall Request → Deleted Deliver Recall/Fallback to traditional recalling → Delivered</p> |
| Notification | Describes whether the sender was notified of the recall result. When a recall request is submitted from Outlook, we have the opportunity to select whether we want to be notified of the outcome. The field can thus take one of the values: <ul style="list-style-type: none"> • Sent • NotSent <p>By default in Outlook the Notification setting is enabled, so most often this value will be set to Sent.</p> |
| Archive | If archiving is enabled, here we get the filename of the archived email. Of course this is only filled if the email was successfully recalled. |

4.4 Status Reporting

Status Reporting enables WinDeveloper Message Recall to alert the administrator of any important error conditions or any other issues that require administrative attention.



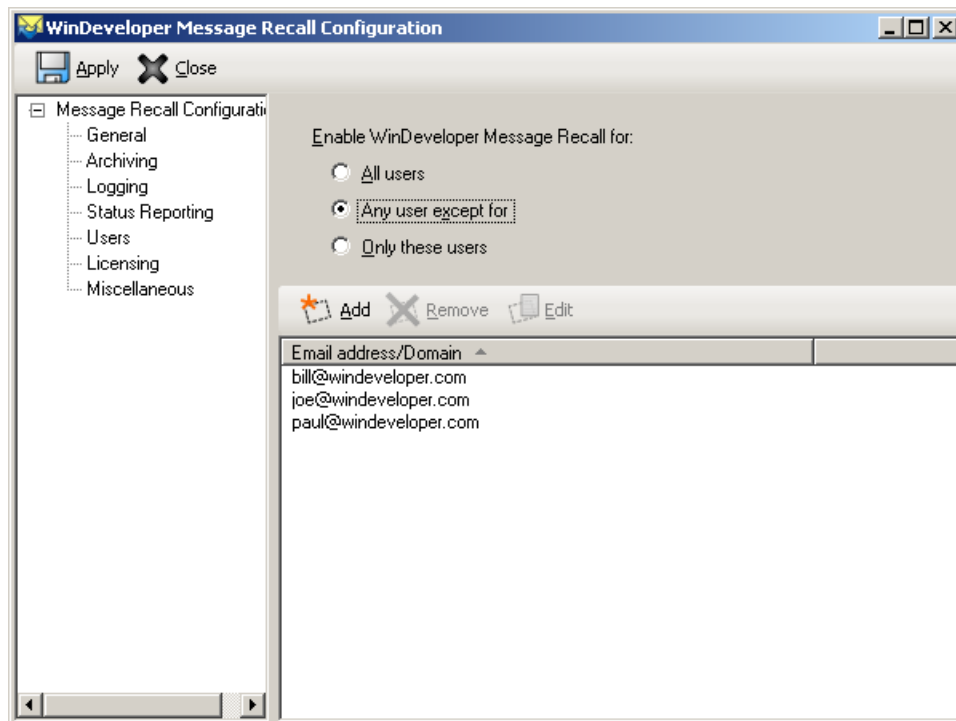
Reports are sent to the administrator via email.

To activate this functionality, set the top most checkbox and enter the email address of the administrator responsible for WinDeveloper Message Recall.

By default both the sender and recipient addresses will be set to the same Administrator address. If you prefer that the application uses a different sender address select **Set the notification sender using this address**, and enter the sender address at the edit box that follows.

4.5 Enable WinDeveloper Message Recall for Individual Users

By default WinDeveloper Message Recall will process all emails submitted by local users.



We can change this at the Users category from the options under **Enable WinDeveloper Message Recall for** as follows.

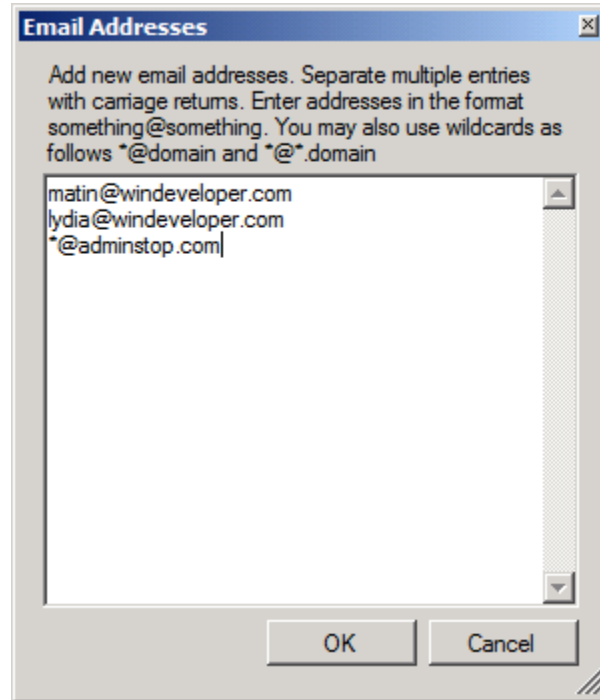
All Users – All local users will have their emails processed.

Any user except for – All local users will have their emails processed with the exceptions of the users included in the list that follows.

Only these users – Only users explicitly included in the list that follows will have their emails processed.

Underneath these three options we find the user list where we can add the email addresses/domains of those users to be included or excluded from processing.

Just click Add to enter a list of user addresses/domains to the list.



Likewise click Delete or Edit to update the user list as necessary. Note how we can enter full email addresses or using the * wildcard we can identify an entire SMTP domain.

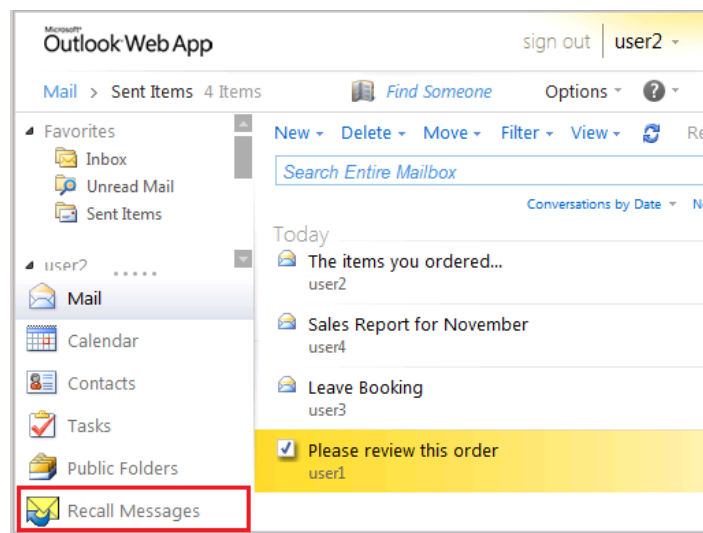
5. Working with WinDeveloper OWA Recall

WinDeveloper Message Recall v2.0 introduced the new OWA Recall extension. This enables users to block delivery of emails sent in error directly from MS Exchange Outlook Web Access/App.

For many years MS Outlook has been providing the 'Recall This Message' command. However this functionality never made it into OWA. WinDeveloper OWA Recall is now bridging this gap, making recalling from OWA even more intuitive than in Outlook.

5.1 OWA Premium

[OWA Recall is installed on each Exchange 2007/2010 Client Access server](#). This adds a new option to the OWA Premium interface sidebar as shown in the screenshots that follow.



OWA Recall extension in Exchange 2010 OWA

Note: All screenshots are taken from Exchange 2010. The same functionality is also available for Exchange 2007.

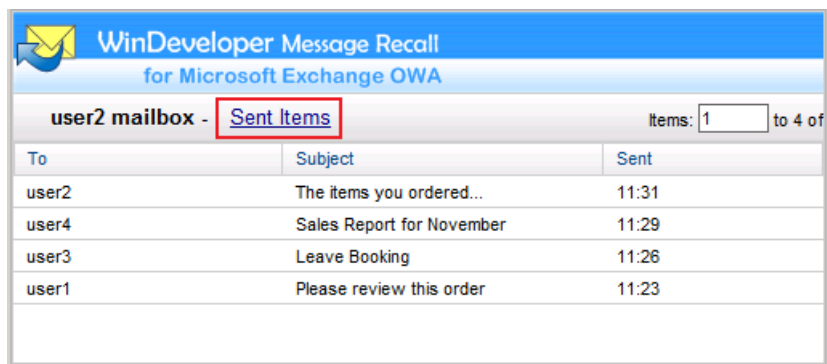
Note: The OWA Recall extension may look a bit different. The way the extension is displayed depends on the number of options present at the sidebar. Once the option count exceeds 6, additional sidebar entries are displayed as small icons.

Note: MS Exchange provides the OWA Premium and Light interfaces. OWA Premium is extensible, whereas OWA Light is not. When logging onto OWA, which of the two interfaces is returned depends on the browser type in use. If working with OWA Light, it is still possible to access OWA Recall by entering the URL:

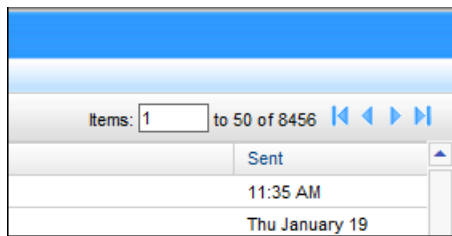
<https://<Client Access server>/owa/forms/owarecall/owarecall.aspx>

5.2 The OWA Recall Interface

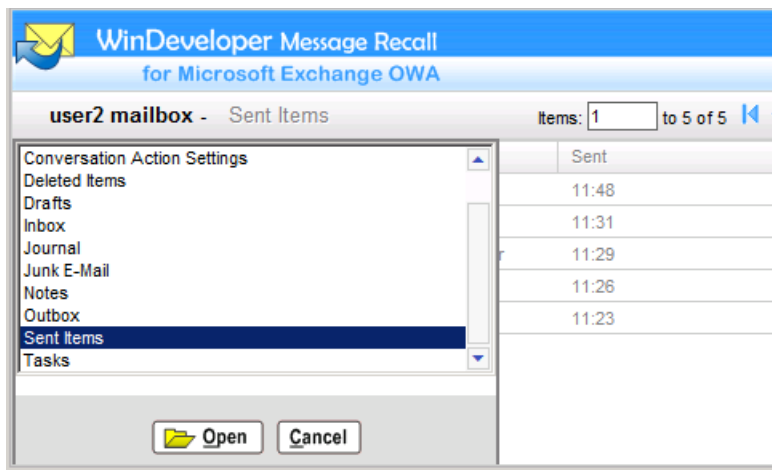
Once OWA Recall is installed click on the sidebar button whenever you want to recall an email. This will open a new browser tab listing the emails under the Sent Items folder.



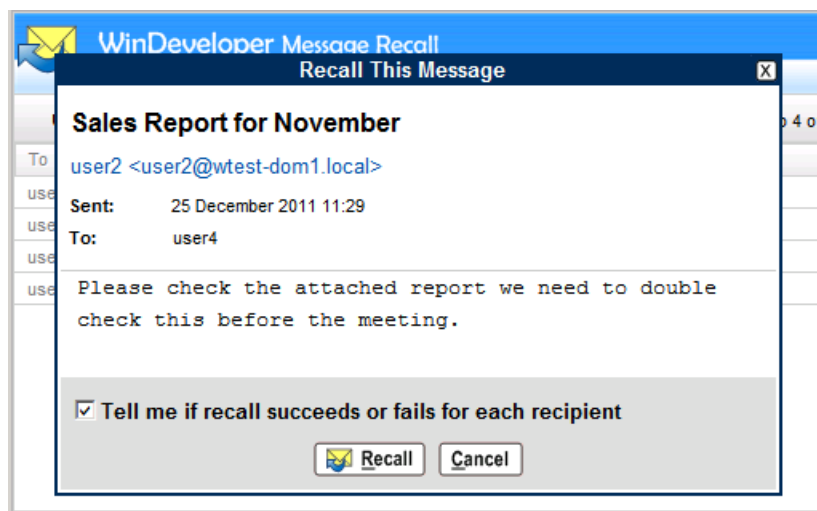
The OWA Recall page shows emails in batches of 50 sorted by date/time with the latest emails on top. If you need to navigate to emails sent earlier, use the navigation arrows shown at the top right corner.



If you need to change the folder, click on the folder name link to open the folder selection list.



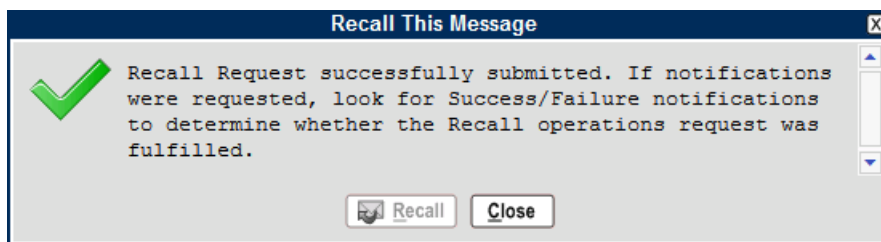
To change the current folder just select a new folder and click Open. Once you identify the email to be recalled, double click the email item.



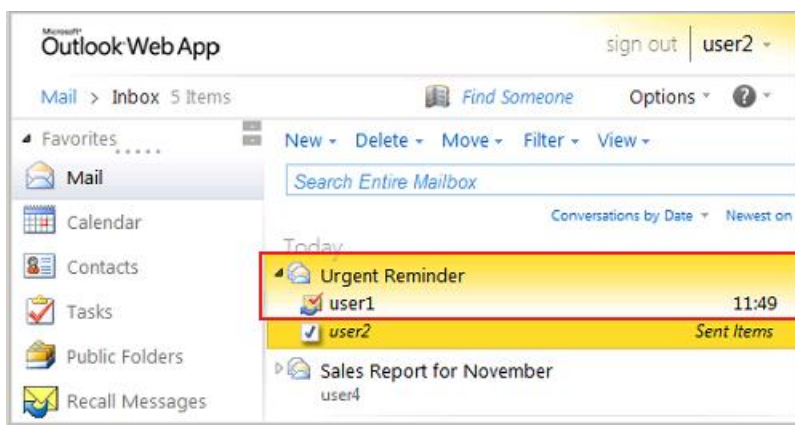
The 'Recall This Message' dialog that opens shows a summary of the email being recalled. At the bottom you have the options for submitting the Recall.

To specify whether notifications should be returned select/clear the checkbox '**Tell me if recall succeeds or fails for each recipient**'.

To submit the recall request, click on the Recall button. This will submit the request and the dialog switches to report whether this was submitted successfully.



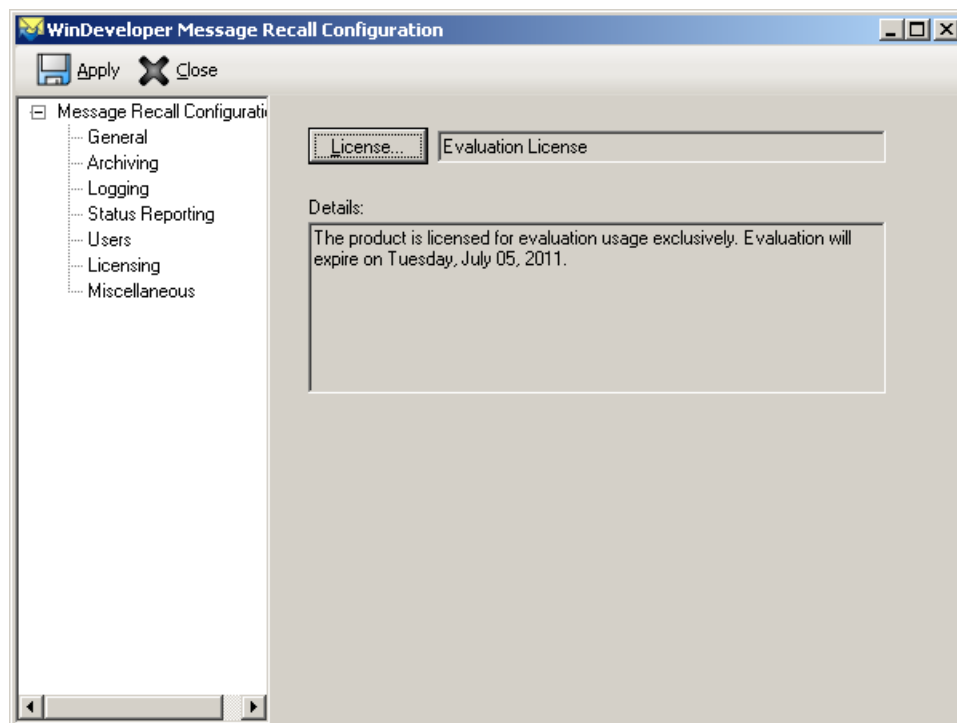
This dialog is only confirming that the recall request was submitted. Whether or not the recall is fulfilled will be reported through notifications (if requested). Notifications will be delivered to the Inbox, so we can close OWA Recall and move back to OWA. This is how a success notification looks like. It is identical to the notifications MS Outlook generates.



6. Licensing WinDeveloper Message Recall

WinDeveloper Message Recall on first install runs in free evaluation for a period of time. Once evaluation is over, the product stops processing emails. At this point a license key must be supplied in order to restore full product functionality.

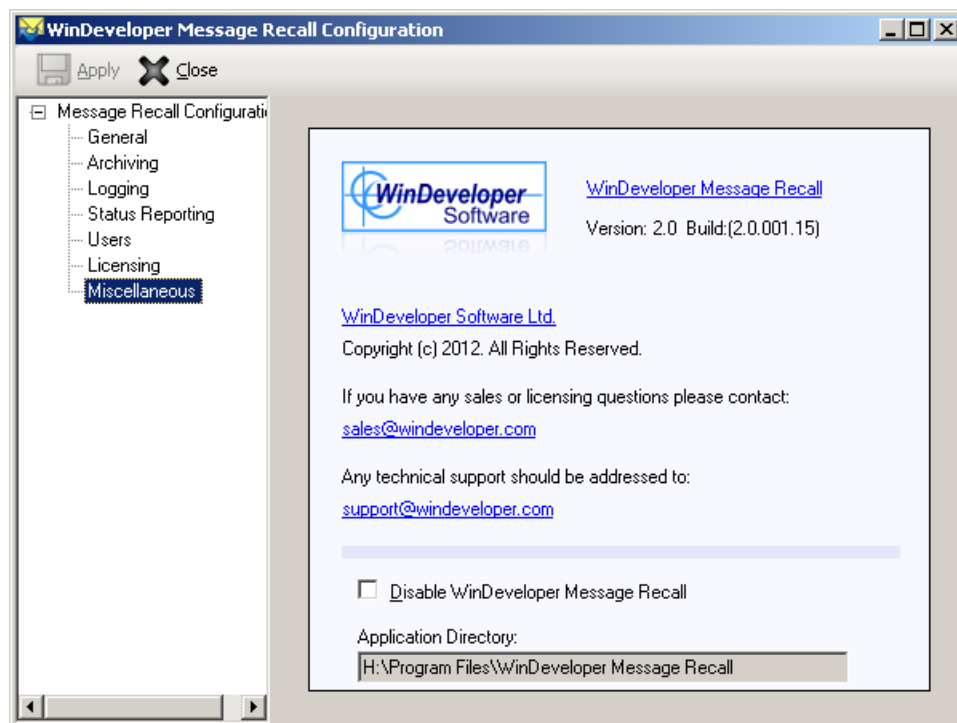
The currently active licensing mode can be verified though the configuration Licensing category. On installing, check out the information at this page. It will show until when the product is licensed for evaluation. In case an extended evaluation period is required email WinDeveloper sales at: sales@windeveloper.com



On ordering a license, the key will be supplied together with step by step instructions on how to license the product. The exact licensing procedure will depend on the type of license key ordered thus it is important to follow the instructions accompanying the key.

7. Contacting WinDeveloper

The Miscellaneous configuration category provides links to the most important contact information.



From here the following links are available:

1. A link to the product homepage:
<http://www.windeveloper.com/recall/>
Check here for the latest information on the product and FAQs.
2. A link to the WinDeveloper Software homepage:
<http://www.windeveloper.com/>
3. Sales/Licensing email address:
sales@windeveloper.com
4. Technical support email address:
support@windeveloper.com

When contacting WinDeveloper please choose the correct email address as outlined above. This will help us provide the quickest response.

If you encounter problems with sending emails go to the WinDeveloper website and use the contact form.