Mailgate Ltd.

MailGate Spam Filter
User Manual

Edited by Lani K. and David D. Thompson.

Contents

Figures

MailGate Spam Filter Extension

Spam Filter Extension Introduction

What Is Spam

In the true sense there are two types of Spam - Unsolicited Commercial Email (UCE) and Unsolicited Bulk Email (UBE). This involves the spammer obtaining your email address and using it to send you mails you have not asked for, often by using means to hide the origins of the message.

For many people though Spam means "any email I did not ask for or do not want". Some of this mail could be regarded as solicited because often it is caused by completing on line registrations or by signing up to list servers. Sometimes getting your address de-listed again can be difficult so you continue to receive un-wanted email.

What is the Spam Filter

The MailGate Spam Filter extension is an optional module which will check all mail passing through MailGate against a set of rules to try to filter out Spam emails.

The filter is designed to identify true Spam by looking for certain characteristics of this type of mail and also can be adjusted to block other un-wanted mails.

Some rules are simple checks on the structure or content of the mail, others make use of pattern and word data lists to support the rule. These lists are referenced by individual rules and are fully user maintainable.

Once a mail has been identified as potential Spam, the module can be configured to perform user configurable actions. The actions taken can be linked to a certainty or priority associated with each rule. This allows an easily identified Spam mail to be treated differently to a mail that is only maybe Spam.

Refer to **How Spam is Filtered** on page 11 for full details of the filtering process.

Installation and Configuration

Spam Filter Installation

To install the MailGate Spam Filter extension, run the self-extracting executable set-up program. This program, after you have read and confirmed your acceptance of the software licence, will install all the necessary components into your MailGate system folder.

Once the install process has completed you must access the extension and configure it to your requirements. To configure the extension, highlight it in the MailGate Main Window and select Edit | Edit or double click on the icon. When you have completed and saved your settings, stop and restart the MailGate service to initialise the extension.

The Spam Filter extension is installed with a full set of pre-defined rules. These are listed in the **Rules Tab** (see page 3) and you should review which rules you wish to use and adjust the priority flags as required.

Some rules reference data lists. These lists are provided with sample data and you should edit them according to your requirements through the **Patterns Tab** (see page 4) and the **Words Tab** (see page 6).

When the Filter identifies a mail as Spam, it will perform the actions defined in the **Action Tab** (see page 8) according to the priority level attached to the mail. Use this tab to define the required actions for each priority level used.

Like all MailGate software, the Spam Filter extension may be tried for 30 days after installation. If you wish to continue using the module you must purchase a key and install this using the Register option in the **About Tab** (see page 10) .

We recommend you read **How Spam is Filtered** on page 11 before finalising your configuration. This will assist in ensuring you get the best from the module.

 If you have more than one MailGate extension installed in your system you should consider the sequence in which the extensions load. Using the best sequence can reduce system resource requirements by keeping unnecessary processing to a minimum. See the MailGate User Manual for more details.

Rules Tab

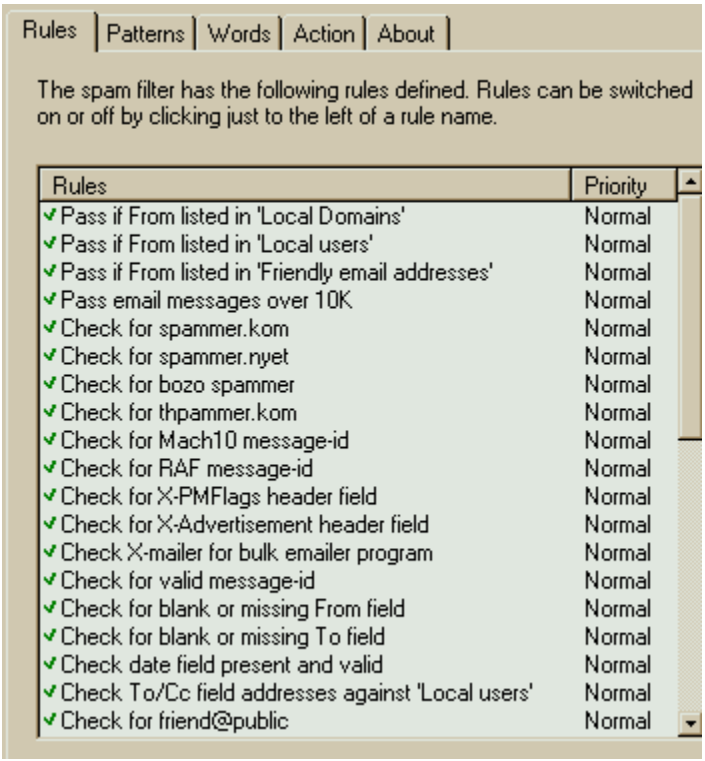

Figure 1 - Rules Tab

The Rules Tab displays a list of the rules used by the Spam Filter when checking the mail traffic. Use the scroll bar to move through the list of rules.

In the Rules List you can :-

- **Enable/Disable a rule**

  Click to the left of the rule. Enabled rules are marked with a green check and disabled rules with a red cross.

- **View the rule details and code**

  Left double click on a rule and a pop-up window will display the details and code for the selected rule. See **About Rules** on page 12 for more details on the rule syntax.

- **Change the rule priority**

  Right click on the current rule priority and a pop-up menu will be displayed. Select the required priority. For more information on the use of priorities see **How Spam is Filtered** on page 11.

Remember to configure actions in the **Action Tab** (see page 8) for all the priorities you are using.

Patterns Tab

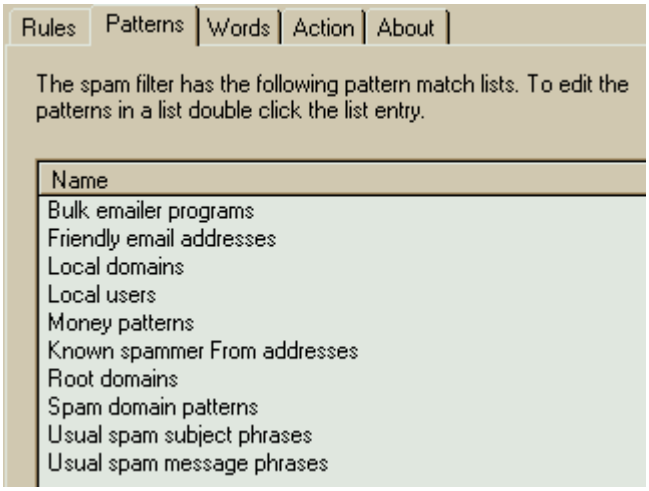

Figure 2 - Patterns Tab

The Patterns Tab displays a list of the installed pattern lists used by individual rules.

The Patterns list box (Figure 2) shows the name given to each list and it is this which is used by the rules to reference the appropriate list.

Double click on an entry to display the **Patterns List Dialog** (see page 5). In this dialog the list contents may be viewed or changed.

Each Pattern list is stored as a file in the MailGate Spam Filter system folder. The names displayed in this tab are the file names used. See **About Patterns** on page 13 for more details.

Patterns List Dialog

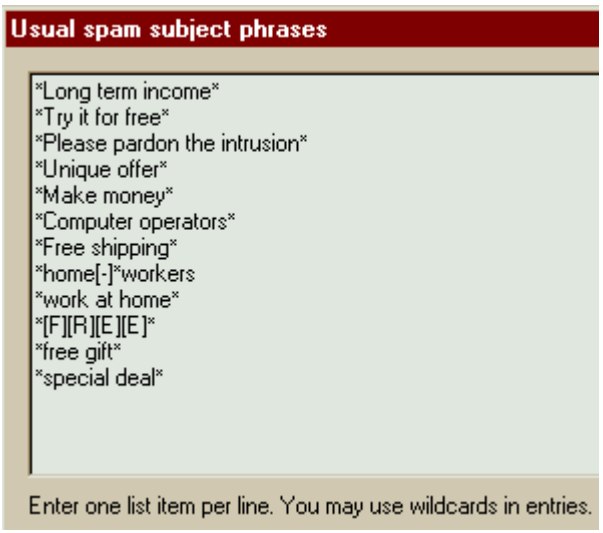

Figure 3 - Example Pattern Dialog

The Patterns List Dialog displays the contents of a Patterns List. To access the dialog, double click on a list name in the **Patterns Tab** (see page 4).

Patterns may be added to the list in any order according to the following rules :-

- Each pattern entry must be placed on a new line.
- The use of wildcards (see the MailGate User Manual Technical Reference section) is allowed.
- Patterns are not case sensitive.
- When adding patterns to a list, remember to check which rules use the list and which parts of the mail are being checked for a pattern match. This context may affect the entries you wish to make.
- When you have finished editing the list, click OK to save your changes.

Words Tab

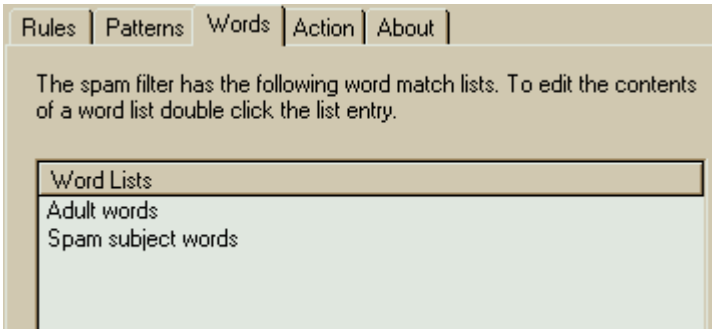

Figure 4 - Words Tab

The Words Tab displays a list of the installed words lists used by individual rules.

The Words list box (Figure 4) shows the name given to each list and it is this which is used by the rules to reference the appropriate list.

Double click on an entry to display the **Words List Dialog** (see page 7). In this dialog the list contents may be viewed or changed.

Each Words list is stored as a file in the MailGate Spam Filter system folder. The names displayed in this tab are the file names used. See **About Words** on page 14 for more details.

Words List Dialog

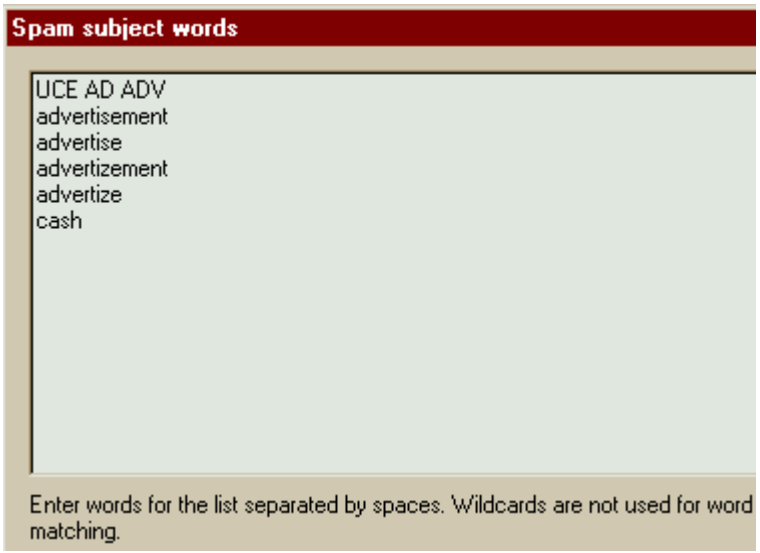

Figure 5 - Example Words List Dialog

The Words List Dialog displays the contents of a Words List. To access the dialog, double click on a list name in the **Words Tab** (see page 6).

Words may be added to the list in any order according to the following rules :-

- Each word entry must be separated by spaces.
- Any number of words may be placed on a line.
- The use of wildcards is not permitted.
- Word matching is not case sensitive.

When adding words to a list, remember to check which rules use the list and which parts of the mail are being checked for these words. This context may affect the entries you wish to make.

When you have finished editing the list, click OK to save your changes.

Action Tab

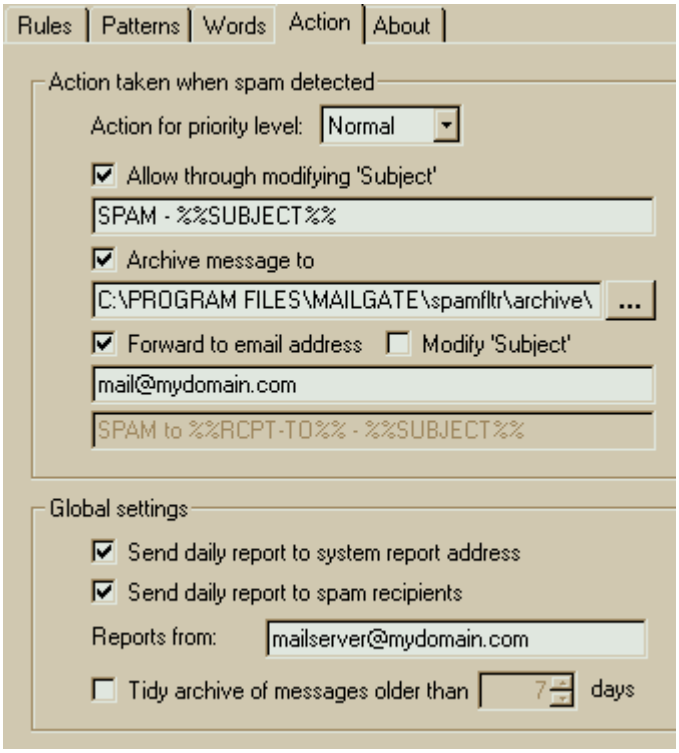

Figure 6 - Action Tab

The Action Tab is used to define the actions to be taken by the Spam Filter Extension when a mail has been identified as Spam. The tab is divided into two sections. The upper section is used to define what happens to a Spam mail. The lower section sets the general operation of the extension.

Action taken when spam is detected

When a mail is checked by the Spam Filter and is matched by one of the rules, a priority level is associated with the message. (See **How Spam is Filtered** on page 11). In this part of the Action Tab you can set what actions you wish to be performed for each priority level.

To set the actions, first select the desired priority level using the **Action for priority level** dropdown (Figure 6). You will now see displayed the current actions for that priority level and you can make any changes required.

For each priority level you can:-

- Have the Spam Filter pass the mail through to its recipient(s) after modifying the subject line, by setting the **Allow through modifying 'Subject'** (Figure 6) option. The subject can be modified using the MailGate Macro facility.

  Use the *Subject Pattern* entry to define how the Spam mail 'Subject' line should be modified.

  When specifying a pattern you can define fixed text or use the MailGate macro facility. These macros allow data from the original mail to be used in the revised 'Subject'.

  Example :-

  A subject pattern setting of "SPAM - %%SUBJECT%%" will add "SPAM - " in front of the original subject.

  For more details of using the macro facility, please refer to the Technical Reference section in the MailGate User Manual.

- Have the Spam Filter Archive a copy of the message using the **Archive message to** (Figure 6) option. The message copy will be added to a daily archive file in the specified folder. The file name format used is **SF<yymmdd>.TXT** where <yymmdd> is the current date.

- Have the Spam Filter forward a copy of the message to another mail address using the **Forward to email address** (Figure 6) option. If using this option, the Filter can also be set to modify the subject line for the forwarded copy using the MailGate Macro facility. If the address is a local mailbox then only the mailbox name is required, but a full email address may be used. This will be required if the mail is to be forwarded externally.

 You should review the actions for each of the priority levels you have associated with the rules in the **Rules Tab** (see page 3).

Global settings

The Global settings section is used to set the general operation of the extension.

By selecting the **Send daily report to system report address** (Figure 6) you can have the Filter send a daily report of Spam message activity to the system administrator email address. This address is the System Reports To address set in the MailGate Setup/Email tab.

Similar daily reports can be sent to each of the Spam mail target recipients by using the **Send daily report to spam recipients** (Figure 6) option.

If you use either of the daily reports options, you should also set the **Reports from** (Figure 6) address. When the daily report email is created, this address is used as the From: address in the mail.

The **Tidy archive messages older than** (Figure 6) option is used to make the extension remove old archive files automatically. Set the *Purge Period* to the number of days which archive files are to be kept for. If the Tidy archive option is checked then files older than this setting will be deleted. If not checked, old archive files will be kept indefinitely.

 In many cases the identification of Spam mail is not an exact science. You should consider making settings that allow you to monitor and review the identification process when you first setup the module and when your rules and list settings are tested you can set your actions with confidence.

About Tab

The About Tab shows the Spam Filter Extension version and copyright details.

This tab also contains the Register button that allows you to enter the extension activation key to licence the module. When you click on this button the Registration dialog will be displayed.

Register Dialog

The Register dialog is accessed by clicking on the Register button in the **About Tab** (page 10).

To register the Spam Extension Module you need to purchase an activation key. Enter this key into the Activation key field EXACTLY as sent to you. All letters should be upper-case.

When you have entered the key, click the OK button to confirm the key is valid and to complete the registration process.

Full details on how to purchase a licence key can be found on our web site at www.mailgate.com.

Using the Spam Filter

How Spam is Filtered

The MailGate Spam Mail Filter Extension checks every mail that passes through the system against a set of rules. This is a matching process where the mail is checked against a set of criteria defined by the rule. The rules may be viewed and adjusted in the **Rules Tab** (see page 3). For details on how the rules work see About Rules on page 12.

Some rules need to use variable data to achieve the desired results. Such rules reference data lists which can be reviewed and edited in the **Patterns** and **Words** tabs (pages 4 and 6). For full details on these lists see **About Patterns** on page 13 and **About Words** on page 14.

There are three possible results after each rule is processed :-

1. The mail is matched by the rule and is identified as spam.
2. The mail is matched by the rule and is identified as not spam.
3. The mail is not matched by the rule.

Should a match be made, then this match is noted along with the rule priority level. Rule processing continues until either all the rules have been tried or the highest achievable priority level has been reached.

Mail messages that fail to be matched by any rules are passed on by the Spam Filter for normal processing.

For those mail messages that have been matched, the status of the highest priority level is checked. If this status indicates the mail is **not** spam then it is also passed on for normal processing. If the status indicates that it **is** spam, then the filter module will apply the actions defined on the **Action Tab** (see page 8) for the mail's priority level.

When you first use the Spam filter, you should take care with the actions you define. It is advisable to run the system for a while carefully checking you are correctly identifying your spam mails to confirm your settings before using actions to block mail deliveries.

Technical Reference

About Rules

The Spam Filter is configured with a standard set of rules. When a mail is passed through the extension each rule is processed in turn until the mail is either matched by a rule or all the rules have been tried. See **How Spam is Filtered** on page 11 for more details.

The rules set can be viewed in the **Rules Tab** (see page 3) and by double clicking on any rule the associated code will be displayed. The rules use an internal scripting language details of which can be found in the MailGate User Manual Technical Reference section under Scripting. The Spam Filter also has a number of special functions (see page 15) available only to the filter rules.

The rules are contained in three simple text files, **userpre.sfr**, **system.sfr** and **userpost.sfr** which are located in the spamfltr folder under the MailGate system folder (by default c:\program files\mailgate). When the rules are loaded by the filter the files are read in the above order.

The extension is shipped with a standard set of rules contained in **system.sfr** which should not be changed. If you wish to create your own rules these should be placed in either **userpre.sfr** or **userpost.sfr**. To add a new rule you may edit these files with a suitable text editor.

The Spam Filter functions are defined using the script language external function directive. The definitions are contained in the file **spamfltr.inc**. This file is read when the rules are loaded and if you wish to create your own functions, these may be included in this file. Otherwise you should not change this file.


Always take a backup copy of your existing files before making any changes. Only edit the files if you are sure you know what you are doing! If you make changes to either of the system files, these must be backed up before re-installing or upgrading the extension module. Your changes may need to be re-done after installing.

Rule Location - You should consider carefully the location of your rule. To minimise system resource requirements, rules which are more likely to find a match should be placed to the top of the rules list by using the file **userpre.sfr**. Rules which are rarely matched may be added to the end of the list by using userpost.sfr. Remember rule processing stops if a high priority match is found.

Rule Structure - Each rule starts with the statement - #rule "Rule Name" - where "Rule Name" is the name displayed in the Rules Tab.

Generally this will be followed by a number of comment lines starting with - //. These are used to describe your rule.

Next comes the rule code itself using the MailGate scripting language.

Finally the rule is terminated with the statement - #endrule

The line // ---------- is simply included to make reading the file easier.

Example

```
#rule "My Example Rule"
//
// This rule will reject any email with 'Spam' in the subject.
//

field$ = HeaderFieldValue("Subject")
if WildcardMatch(field$,"Spam") then
  IsSpam()
endif

#endrule
//--------------------------------------------------------------------------------
```

In this example the content of the mail header field 'Subject' is obtained and then it is checked for the word "Spam". If this is found then the mail is marked as being a Spam mail.

Note - When you have completed you changes to the rules file you must stop and re-start the MailGate service to register your changes. If you are making many changes at one time, you should work on an off-line copy of the file.

About Patterns

Spam Filter rules can make use of data lists which may be accessed using either the Patterns or the Words Tabs.

In the **Patterns Tab** (see page 4) you will find a list of all the Patterns lists installed.

Each pattern list is contained in a simple text file called <ListName>.lst which is located in the spamfltr folder under the MailGate system folder (by default c:\program files\mailgate). <ListName> is the name you see in the Patterns Tab and also is used by the rules to identify the list.

To add a new list, create a text file and give it a name as above. You will need to close and re-open the extension configuration screen for your new list to appear in the Patterns Tab.

List Contents

You can edit a list's contents either through the Patterns Tab or by using your text editor.

Patterns may be added to the list in any order according to the following rules :-

- Each Pattern must be placed on a new line.
- The use of wildcards (see the MailGate User Manual Technical Reference section) is allowed.
- Patterns are not case sensitive.

About Words

Spam Filter rules can make use of data lists which may be accessed using either the Patterns or the Words Tabs.

In the Words Tab (see page 6) you will find a list of all the Words lists installed.

Each words list is contained in a simple text file called <ListName>.wrd which is located in the spamfltr folder under the MailGate system folder (by default c:\program files\mailgate). <ListName> is the name you see in the Words Tab and also is used by the rules to identify the list.

To add a new list, create a text file and give it a name as above. You will need to close and re-open the extension configuration screen for your new list to appear in the Words Tab.

List Contents

You can edit a list's contents either through the Words Tab or by using your text editor.

Words may be added to the list in any order according to the following rules :-

- Each word entry must be separated by spaces.
- Any number of words may placed on a line.
- The use of wildcards is not permitted.
- Word matching is not case sensitive.

Spam Filter Script Functions

Script Functions for Spam

The Spam Filter extension makes use of the MailGate scripting language. For full details of the use of this language refer to the MailGate User Manual Technical Reference section on scripting.

In addition to the standard internal scripting functions, the Spam Filter extension makes available a number of scripting functions specific to the operation of the extension.

These functions are defined automatically using the external function directive and are available to all the Spam Filter rules. The available functions are listed below with full details on the following pages.

Spam Filter Functions

IsOK()
IsSpam()
HeaderFieldExists(<FieldName>)
HeaderFieldValue(<FieldName>)
WildcardMatch(<String>,<MatchPattern>)
MatchesListItem(<PatternList>,<String>)
ParseAddress(<AddressString>)
MessageSize()
FindWordInString(<WordList>,<String>)
FindWordInMessage(<WordList>)
GetFirstAddress(<FieldName>)
GetNextAddress()
IsValidDate(<DateString>)
WildcardMatchHeader(<MatchPattern>)
WildcardMatchBody(<MatchPattern>)
BodyMatchesListItem(<PatternList>)

Function IsOK()

Parameters - None

Return Type - None

Operation
Calling this function identifies the current mail as **NOT** Spam and registers this along with the current rule's priority level.

Function IsSpam()

Parameters - None

Return Type - None

Operation
Calling this function identifies the current mail as Spam and registers this along with the current rule's priority level.

Function HeaderFieldExists(<FieldName>)

Parameters -

Value	Type	Description
<FieldName>	string	The header field to look. Do not include the ':'

Return Type - Logical Integer

Operation
Looks for the specified mail header field. Returns True if the field exists otherwise returns False.

Example

```
if HeaderFieldExists("From") then
  <action>
endif
```

<Action> is performed if the From: field is found in the mail header.

Function HeaderFieldValue(<FieldName>)

Parameters -

Value	Type	Description
<FieldName>	string	The header field whose value is required. Do not include the ':'

Return Type - String

Operation
Returns the value associated with the specified header field.

Example
```
if HeaderFieldValue("From") = "Spammer" then
  IsSpam()
endif
```

If the From: field in the mail header exactly matches the string "Spammer" then the mail is identified as Spam.

Function WildcardMatch(<String>,<MatchPattern>)

Parameters -

Value	Type	Description
<String>	string	The string to search for a pattern match.
<MatchPattern>	string	The wildcard pattern string to search <string1> for.

Return Type - Logical Integer

Operation
Checks <string 1> for a match to <string2> where <string2> can contain wildcard specifiers. Returns True if a match is found otherwise returns False.

Example
`WildcardMatch("ABCDEF","*BCD*")` will return True.
`WildcardMatch("ABCDEF","*BDE*")` will return False.

Function MatchesListItem(<PatternList>,<String>)

Parameters -

Value	Type	Description
<PatternList>	string	The name of the patterns list to use as a string.
<String>	string	The string to be checked for a pattern match.

Return Type - Logical Integer

Operation
Check for a match between <string2> and any of the items in the specified patterns list. Returns True if a match is found otherwise returns False.

Example
```
MatchesListItem("My Patterns","String to be Checked")
```

This will return True if there is a pattern in the list "My patterns" which matches to the string "String to be checked". Note that wildcards can be used in the Patterns lists and matching is not case sensitive.

Function ParseAddress(<AddressString>)

Parameters -

Value	Type	Description
<AddressString>	string	String from which to extract a valid email address.

Return Type - String

Operation
Extracts a valid email address from the specified string. In many cases an email header field will contain a name along with the email address.

Example
`ParseAddress("(Doe, John) <jdoe@domain.com>")`
returns the string `jdoe@domain.com`

Function MessageSize()

Parameters - None

Return Type - Integer

Operation
Returns the size of the current message in bytes.

Function FindWordInString(<WordList>,<String>)

Parameters -

Value	Type	Description
<WordList>	string	The words list to use as a string.
<String>	string	The string to check for a word match.

Return Type - Logical Integer

Operation
Checks the words in <string2> for a match against all the words contained in the specified words list. Returns True if a match is found otherwise returns False.

Note that wildcards are not permitted in the Words lists so words must exactly match but matching is not case sensitive.

Example
`FindWordInString("My Word List","I am a subject")`

This will return True if any of the words in "I am a subject" exist in the words list "My Word List".

Function FindWordInMessage(<WordList>)

Parameters -

Value	Type	Description
<WordList>	string	The words list to use as a string.

Return Type - Logical Integer

Operation
Checks the words in the entire message for a match against all the words contained in the specified words list. Returns True if a match is found otherwise returns False.

Note that wildcards are not permitted in the Words lists so words must exactly match but matching is not case sensitive.

Function GetFirstAddress(<FieldName>)

Parameters -

Value	Type	Description
<FieldName>	string	The header field to extract the first address from. Do not include the ':'

Return Type - String

Operation
Some mail header fields (particularly the To: and CC: fields) can contain multiple addresses. This function extracts the first valid email address from the specified field.

Example
```
GetFirstAddress("To")
```
will return the first valid address found in the To: header field. See GetNextAddress() for reading multiple addresses.

Function GetNextAddress()

Parameters - None

Return Type - String

Operation
Special function used in conjunction with GetFirstAddress() to enable reading of multiple addresses from the To: and CC: header fields.

Example
```
addr$ = GetFirstAddress("To")
:loop
    if length(addr$) = 0 then loopend:
    if <test the address> then goto notspam:
    addr$ = GetNextAddress()
    goto loop:
:loopend
IsSpam()

:notspam
```

This example demonstrates the reading of multiple addresses from the To: header field. Using the label jump capability, a loop is defined. The first address is extracted

and the loop entered. An exit from the loop will occur either if the extracted address is 0 length (i.e. there is no address) or the address tests as Not Spam. After the first address has been tested, the next available address is extracted by GetNextAddress() and the looping will continue until GetNextAddress() returns an empty string or an address tests as Not Spam.

Note that GetNextAddress() requires GetFirstAddress() to define the header field to read.

Function IsValidDate(<DateString>)

Parameters -

Value	Type	Description
<DateString>	string	String to check for a valid date. Typically this will be the Date: mail header field.

Return Type - Logical Integer

Operation
Checks the specified string for characters which represent a valid date. Returns True if a valid date is found otherwise returns False.

Example
```
IsValidDate("Fri, 6 Oct 2000 09:05 +0100 (BST)")
```
will return True.

Function WildcardMatchHeader(<MatchPattern>)

Parameters -

Value	Type	Description
<MatchPattern>	string	The wildcard pattern to look for in the mail header

Return Type - Logical Integer

Operation
Checks the entire mail header for a match against the specified wildcard pattern. Returns True if a match is found otherwise returns False.

Example
```
if WildcardMatchHeader("*@spammer.kom*") then
   IsSpam()
endif
```
Identifies the mail as being Spam if "@spammer.kom" is found anywhere in the header.

Function WildcardMatchBody(<MatchPattern>)

Parameters -

Value	Type	Description
<MatchPattern>	string	The wildcard pattern to look for in the mail body.

Return Type - Logical Integer

Operation
Checks the entire mail body for a match against the specified wildcard pattern. Returns True if a match is found otherwise returns False.

Example

```
if WildcardMatchBody("*make millions*") then
   IsSpam()
endif
```

Identifies the mail as being Spam if the phrase "make millions" is found anywhere in the mail body.

Function BodyMatchesListItem(<PatternList>)

Parameters -

Value	Type	Description
<PatternList>	string	The name of the patterns list whose items are to be checked against the message body.

Return Type - Logical Integer

Operation
Check for a match between each of the items in the specified Patterns list and the content of the message body. Returns True if any list item matches otherwise returns False.

Example

```
if BodyMatchesListItem("Usual spam message phrases") then
   IsSpam()
endif
```

Message is marked as Spam if a pattern match is found between any of the items in the Patterns list "Usual spam message phrases" and the message body content.

Spam Filter Windows Registry

Windows Registry

The Spam Filter extension stores its configuration in the Windows Registry. All settings are stored under the section:-

HKEY_LOCAL_MACHINE

Software

IDSL

MailGate

SpamFltr

Within the SpamFltr key are found the main configuration settings. There is also a subkey named Priority. This key has an entry for each rule and the value is the rule priority level as defined in the **Rules Tab** (see page 3). See the following pages for details.

Registry - SpamFltr

Most of the Spam Filter extension configuration settings are stored in this Registry key.

Some of the settings are in the form <SettingName><n> where n is the value of the priority for an action setting. n is in the range 1 to 5 where 1 = highest priority and 5 = lowest.

Value Name (Data Type)	Description
Archive<n> (DWORD)	Indicates whether the action should archive a copy of a spam message. 1=Archive Copy. Default=0.
ArchivePath<n> (String)	Provides the drive and path to use when archiving a spam message
DisableRules (Multi-String)	An array of the names of all rules that are disabled.
Forward<n> (DWORD)	Indicates whether the action should forward a copy of a spam message to a given address. 1=Forward Copy. Default=0.
ForwardAddress<n> (String)	The email address to forward a copy of a spam message to.
ForwardMacro<n> (String)	The macro to use to modify the subject line for a forwarded spam message.
ForwardSubject<n> (DWORD)	Indicates whether the subject of a forwarded spam message should be modified by the macro string. 1=Modify. Default=0.
LastReportDay (DWORD)	Used internally to control the daily reports. Contains the Julian day number for the last report.
LicenceString (String)	When the extension is registered contains the registration key.
ReportAddress (String)	The email address used as the From: address for the daily reports mails.
Subject<n> (DWORD)	Indicates whether a spam message should be passed to the recipient with a modified subject line. 1=Pass through. Default=1.
SubjectMacro<n> (String)	The macro to use to modify the subject line for a spam message passed through to the recipient.
SystemReport (DWORD)	Indicates if a daily system report should be generated. 1=Generate report. Default=1.
Tidy (DWORD)	Indicates if old archive files should be automatically deleted. 1=Delete. Default=0.
TidyDays (DWORD)	The number of days worth of archives to keep when tidying. Default=7.
UserReport (DWORD)	Indicates if a daily user report should be generated. 1=Generate report. Default=1.
Version (String)	Gives the version number of the currently installed extension.

Registry - Priority

The Registry Key has a value entry for each rule listed in the **Rules Tab** (see page 3). The rule name is used for the value name and each is of type DWORD. The DWORD value gives the rule priority in the range 1 to 5 where 1 = highest priority and 5 = lowest.