CA DataMinder

Message Server Integration Guide

Release 14.6

CA Technologies Product References

This document references the following CA Technologies products:

- CA DataMinder™

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- Configure Exchange Agent to Use RPC over HTTP(S)—Recent architectural changes in Microsoft Exchange Server 2013 require you to change your MAPI connectivity configuration.

Contents

Chapter 1: About Message Server Integration

This guide shows how to deploy email server agents to integrate CA DataMinder with Exchange Server, Lotus Domino, IIS SMTP servers, and Sendmail or Postfix.

Important: By default, email server agents are disabled when they are first deployed. After configuration, you must turn on the agents to enable email processing.

This section contains the following topics:

Introduction

Integration with Microsoft Exchange Server, IIS SMTP, Lotus Domino, Sendmail, and Postfix allows CA DataMinder to monitor and control corporate email activity that would otherwise be missed by endpoint integration alone. This includes emails sent using BlackBerry handhelds, Microsoft Office Outlook Web Access or Notes Web Clients.

This section focuses on how to configure the server agents and how to monitor agent activity. It highlights the key differences between client-side and server-side email integration.

Note: Throughout this section:

- 'Exchange Server' (with an upper-case S) always means Microsoft Exchange Server. Conversely, 'Exchange server' (with a lower-case s) means the actual computer hosting:
  - Exchange 2003, or
  - A Hub Transport server for Exchange 2007 or 2010, or
  - A Mailbox server for Exchange 2013
- 'Domino' means IBM Lotus Domino Server. The term 'Domino server' (with a lower-case s) means the actual computer hosting Domino.

Server-side versus Client-side

CA DataMinder supports both client-side and server-side email integration. Specifically, it can integrate with Microsoft Outlook or Lotus Notes on client machines and with Microsoft Exchange, IIS SMTP, Domino, Sendmail and Postfix email servers. But there are two key differences between these email integration methods:

First and most important, the range of available control interventions is slightly more limited under server-side integration. Specifically:

- CA DataMinder can intervene by blocking an email and sending a notification message to the sender; or in the case of emails that generate warning or inform events, it can simply send a notification message to the sender. However, it cannot display warning dialogs that require user interaction (unlike under client-side integration, when CA DataMinder *can* display dialogs that require user interaction).
- With Microsoft Exchange server or IIS SMTP agents, CA DataMinder can also intervene by sending an interactive warning email to the sender. The user can then reply to the message to disregard the warning, or do nothing to automatically heed the warning and the agent sends or does not send the email accordingly. This is possible for emails that generated warning or inform events and requires specific configuration on the machine hosting the agent.

Second, under server-side integration CA DataMinder always applies triggers from the sender's perspective (that is, it applies triggers for outgoing emails). This approach avoids unnecessary duplication of analysis and processing. These triggers are defined in the sender's user policy or, if the sender is not a recognized CA DataMinder user, in the policies for the Unknown Internal Sender or External Sender. This contrasts with client-side integration where CA DataMinder can apply both incoming email triggers (defined in the recipient's policy) and outgoing email triggers (defined in the sender's policy).

About the IIS SMTP Agent

The IIS SMTP agent, WgnSMTPS.dll, enables CA DataMinder to monitor and control emails transiting through Microsoft IIS SMTP servers. These servers typically operate at the Internet boundary. The IIS SMTP agent enables CA DataMinder to analyze emails leaving the company or arriving from an external source.

Automatic Notifications and Email Server Agents

When CA DataMinder intercepts an email passing through an email server, it can generate notification emails. You can specify a global sender, a subject, and the body text for these notification emails.

When does CA DataMinder send notification emails?

The handling for notification emails depends on the type of email server:

Exchange

If the email generates:

- A blocking event, the Exchange server agent sends a plain notification email to the sender.
- A warning or inform event, the Exchange server agent can send a plain notification email or an interactive warning email to the sender.

  Additional agent configuration is needed to enable interactive warning emails.

Domino

If the email generates a blocking, warning, or inform event, the Domino server agent sends a plain notification email to the sender.

IIS SMTP

If the email generates:

- A blocking event, the IIS SMTP agent sends a plain notification email to the sender.
- A warning or inform event, the IIS SMTP agent can send a plain notification email or an interactive warning email to the sender.

  Additional agent configuration is needed to enable interactive warning emails.

Sendmail or Postfix

If the email generates a blocking, warning, or inform event, the Milter MTA agent sends a plain notification email to the sender.

More information:

Global Sender (see page 12)
Subject Text (see page 12)
Body Text (see page 12)

Global Sender

When a user receives a notification email from CA DataMinder, the From: field indicates the sender of the notification email. However, the sender's identity is configurable. For example, you can specify that the From: field in notification emails is always set to ComplianceTeam@Unipraxis.com.

To define a 'global sender', you must edit the registry on the machine hosting the Exchange Server or Domino server agent. Specifically, you need to add the NotificationFromAddress registry value.

Subject Text

You can define the subject for a notification email in the sender's policy. To do this, you edit settings in the System Settings/User Notifications folder:

Blockings

For blocked emails, you define the subject text in the 'Dialog Title - Blockings' setting.

Warnings

For emails that generate warnings, you define the subject text in the 'Dialog Title - Warnings and Quarantine Events' setting.

Inform

For emails that generate inform events, you define the subject text in the 'Dialog Title - Inform, Notify and Categorize Events' setting.

Body Text

You can define the body text for a notification email in the sender's policy. To do this, you edit the Message to Users setting in the relevant control trigger. For example, if an email is blocked when the Search Text 1 trigger activates, the notification email includes this trigger's Message to Users setting as its body text.

This means that for each outgoing email control trigger in a sender's policy, the Message to Users text must reflect the Intervention option specified in the associated control action. For example:

"Your email has been blocked. It refers to %Keystring% and such references violate corporate guidelines. Please contact the Compliance Officer for further details."

Chapter 2: Exchange, Domino and IIS SMTP Integration

This section describes how to deploy the Exchange, Domino and IIS SMTP email server agents.

This section contains the following topics:

Requirements for the Exchange Server Agent

Note the requirements for the Exchange Server Agent:

Supported Versions

Microsoft Exchange Server 2003, 2007, 2010, or 2013

Requirements for Exchange 2013

Install the Exchange 2013 server agent on each mailbox server.

CA DataMinder can analyze and apply policy to emails passing through Microsoft Exchange Server 2013. But see the support limitations in the Announcements section of the *Release Notes*.

Requirements for Exchange 2007 and 2010

Install the Exchange 2007 or 2010 server agent on each Hub Transport Server in the Active Directory site.

See also the 'MAPI client and CDO 1.2.1' requirement.

MAPI client and CDO 1.2.1

CA DataMinder integration with Exchange 2007 and 2010 requires the Messaging API and Collaboration Data Objects 1.2.1 component.

This software is typically referred to as the 'MAPI client and CDO 1.2.1' component. Download the 'MAPI client and CDO 1.2.1' component from the Microsoft Web site. We recommend using the latest build, which is unrelated to the 1.2.1 version number.

Requirements for the Domino Server Agent

Note the requirements for the Domino Server Agent:

Supported Versions

Lotus Domino 7.0.2, 8 or 8.5

CA DataMinder only supports 32-bit versions of these releases. 64-bit versions are not currently supported.

Note: Integration has been tested using the Domino versions listed above. It may work with other versions, but these have not been tested.

Requirements for the IIS SMTP Server Agent

Note the requirements for the IIS SMTP Agent

Microsoft Internet Information Services (IIS)

The SMTP server hosting the IIS SMTP agent must be running IIS 6.0.

32-bit Systems Only

The IIS SMTP agent is only supported on 32-bit systems.

MAPI client and CDO

CA DataMinder integration with IIS requires the Messaging API and Collaboration Data Objects 1.2.1 component.

This software is typically referred to as the 'MAPI client and CDO 1.2.1' component. Download the 'MAPI client and CDO 1.2.1' component from the Microsoft Web site. We recommend using the latest build, which is unrelated to the 1.2.1 version number.

How to Deploy the Exchange, Domino or IIS SMTP Agent

This section describes how to install and configure the Exchange server agent, IIS SMTP agent, or Domino server agent and the policy engine hub.

The key deployment tasks are:

1. **Deploy your policy engines.**

   We recommend you do this before deploying your email server agent. This is described in Deploy policy engines.

2. **Deploy the email server agent and policy engine hub**.

   When you install the email server agent, a policy engine hub is also installed automatically.

   You must then separately configure the agent and hub.

3. **Turn on email server integration.**

   To enable the agent and turn on integration, you edit a registry value on the email server.

   Important! You must enable CA DataMinder email server agents in order to start email processing. By default, the agents are disabled when first deployed.

Where Do I Install an Exchange Server Agent?

Recent releases of Exchange Server are modular applications comprising multiple server roles. Take care to install the Exchange server agent on the correct server!

Exchange Server 2013

The number of server roles in Exchange 2013 was reduced to two. These roles are the Mailbox Server role (which includes the Hub Transport service) and the Client Access Server role.

To enable CA DataMinder to monitor and control all communications sent via Exchange 2013, you must install the CA DataMinder Exchange server agent on each Mailbox server.

Note: For maximum security and performance, follow the best practices that you find on the Microsoft Windows Dev Center under "RPC over HTTP Deployment Recommendations". See the Event Report Requirements section in the *Archive Integration Guide* for details on Exchange and Outlook Import Requirements.

CA DataMinder releases before r14.5 SP1 connected to mailboxes on Exchange Server using RPC. With the architecture changes in Exchange Server 2013, RPC connections are no longer accepted by Exchange Server 2013. You get an error message for a network error or an inaccessible location.

Make all requests to Exchange Server 2013 through **RPC over HTTP** (ROH). Exchange Server 2013 is no longer identified with its hostname like in earlier versions, but with a unique global identifier, the GUID. Therefore, CA DataMinder utilities need to refer to the Exchange Server 2013 by its Exchange GUID along with its domain name. To address this new architecture, you must provide CA DataMinder utilities with additional details.

- Proxy server name
- HTTP authentication method
- RPC authentication method
- Valid SSL certificate

Exchange Server 2007 and 2010

These applications have five server roles. For large organizations, the Exchange environment may be highly complex and can contain multiple Active Directory sites, each of which can contain multiple Mailbox servers and one or more Hub Transport servers. Depending on the needs of your organization, you can install a Mailbox server and a Hub Transport sever on the same computer or on separate computers.

To enable CA DataMinder to monitor and control all communications sent via Exchange 2007 or 2010, you must install the CA DataMinder Exchange server agent on each Hub Transport server.

Deploy Policy Engines

For installation and configuration instructions, see the Policy Engines chapter in the *Platform Deployment Guide*.

In particular read the instructions for specifying the PE domain user. You must specify this user account when you install a policy engine hub.

Install an Exchange, Domino or IIS SMTP Agent

You install the Exchange, IIS SMTP, and Domino server agents using the CA DataMinder Integration Agents installation wizard. A policy engine hub is also installed automatically when you install the server agent.

To install an Exchange, IIS SMTP, or Domino server agent

1. Find setup.exe in the root of your CA DataMinder distribution image. Run setup.exe to launch the CA DataMinder installation wizard.

   The Installation Type screen opens.

2. Click Advanced Installation.
3. In the Advanced Install Options screen, choose Server Agents and then click Install.

   This launches the CA DataMinder Integration Agents installation wizard in a separate window.

4. In the Integration Agents installation wizard, navigate to the Customer Setup screen.
5. In the Custom Setup screen, expand the Server Agents branch and choose Exchange Server Agent, IIS SMTP Agent, or Domino Server Agent.
6. In the Policy Engine Hub Configuration screen, provide the credentials for the PE domain user.
7. In the final wizard screen, click Install to start the file transfer.

   You now need to configure the email server agent and policy engine hub.

Configure the PE Hub

This is described in Configure the policy engine hub (see page 98).

Configure the Exchange, Domino or IIS SMTP Agent

You configure the Exchange server agent, IIS SMTP agent, or Domino server agent by modifying the registry. For Domino servers, you also need to configure their MIME settings.

Clearly, any changes to mission-critical systems such as an email server must be thoroughly tested first. For this reason, CA DataMinder also supports various registry values which, although not installed automatically, can be used for diagnostic purposes.

MIME configuration

(Domino servers only) To ensure that the 'policy processed' status is preserved when emails already processed by CA DataMinder are sent between Domino servers over SMTP links, you need to change the MIME settings.

Modify the registry

To configure the email server agent, you modify registry values (see page 22). For example, you can tailor the email server agent to only monitor emails sent from particular SMTP addresses (such as addresses ending with '@unipraxis.com'). Other registry values determine how the server agent handles event failures and out-of-memory failures. The Exchange server agent and IIS SMTP agent can also be configured to send interactive warning messages.

More information:

Email Server Agent Registry Values (see page 22)
MIME Configuration for Domino Servers (see page 19)

MIME Configuration for Domino Servers

Before enabling CA DataMinder integration with Domino, you need to edit the MIME configuration settings on your Domino servers. This ensures that any emails which have already been processed by CA DataMinder:

- Do not lose their 'policy processed' status when they are sent between Domino servers that send messages using SMTP rather than the NRPC (Notes Remote Procedure Call) protocol.

  Without this configuration change, if a Domino server agent were running on the recipient server this loss of status would mean that the email was processed by CA DataMinder again and, potentially, if the recipient Domino server agent were parented to the same CMS as the originating CA DataMinder client or server agent, a duplicate event would be written to the CMS database.

- Retain any x-headers generated by CA DataMinder. Without this configuration change, these x-headers would be lost from the email. For example, if you use such x-headers to flag emails that must be encrypted, this information would be unavailable to the third party encryption solution.

Edit the Domino MIME options

To edit the MIME configuration settings

1. In Domino Administrator, under Server Configuration you need to edit the advanced MIME configuration settings. To do this, navigate to the MIME tab, then go to the Advanced > Advanced Outbound Message Options tab—see the screenshot below.
2. **To retain the 'policy processed' status:** This status is appended to emails as a Notes private item. You can configure Domino to send all private items or you can explicitly specify the CA DataMinder items. You need to edit one or both of these options:

   - 'Internet Mail server sends Notes private items in messages'. Set this to 'Enabled'. Note that this change alone will not retain CA DataMinder x-headers (see step 3).

     Alternatively, if you are reluctant to make this configuration change (for example, because it may consume excessive bandwidth or because there may be security implications), we recommend you edit the following option instead.

   - 'Always send the following Notes items in headers'. Add these items as a comma-separated list:

     ```
     WiganStatus,WiganSS
     ```

3. **To retain CA DataMinder-generated x-headers:** CA DataMinder can insert x-headers into e-mails through the use of special smart tags in policy triggers. To ensure these x-headers are retained when the emails transit through Domino servers, edit the **Always send the following Notes items in headers** option.

   Add the CA DataMinder-generated x-header, minus its **x—** prefix, to this list (x-headers are temporarily stripped of this prefix when processed by Domino). If appending the x-header to the items you added in step 2**,** use a comma-separated list. For example:

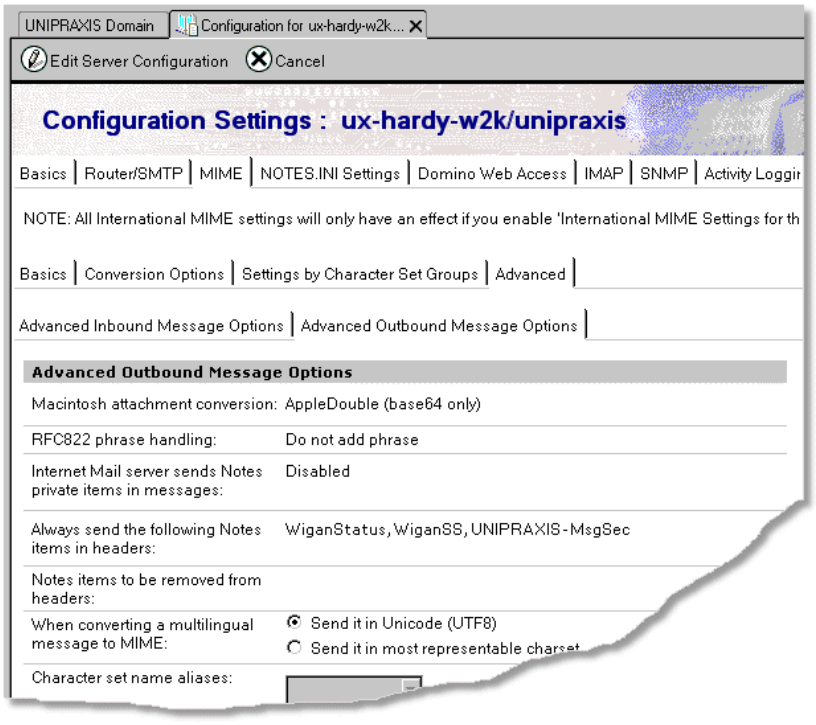
   If the x-header name generated by CA DataMinder smart tags is:

   `X-UNIPRAXIS-MsgSec`

   Then you need to add these Notes items to the list:

   `WiganStatus,WiganSS,UNIPRAXIS-MsgSec`

   If you subsequently change this x-header, or start using additional x-headers, you must edit this list accordingly.



Domino Administrator, MIME Configuration options

Configure the IIS SMTP Agent

Configuration for the IIS SMTP agent is very similar to that for the Exchange server agent. To configure the agent, you modify values in this registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
    \CurrentVersion\Exchange
```

For example, you can tailor the IIS SMTP agent to only monitor emails sent from particular SMTP addresses (such as addresses ending with '@unipraxis.com'). Other registry values determine how the agent handles event failures and out-of-memory failures. The IIS SMTP agent can also be configured to send interactive warning messages.

Further registry changes are needed if you are using the IIS SMTP agent to host a CA DataMinder service for use by other organizations.

More information:

Email Server Agent Registry Values (see page 22)
Hosting a CA DataMinder Service for Use by Other Organizations (see page 51)

Email Server Agent Registry Values

This section lists the available registry values for the email server agent. Some registry values are created automatically when you install the email server agent. Others, primarily the diagnostic values, must be created manually.

Unless explicitly stated otherwise in their individual descriptions, all registry values apply to the Exchange, IIS SMTP, and Domino server agents.Values are located in the following registry keys:

Exchange and IIS SMTP integration

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder\CurrentVersion\Exchange

Domino integration

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder\CurrentVersion\Domino

Email server agents support the following registry values:

- Automatic registry values (see page 23)
  Agent
  AttachOriginalEmail
  EMailFailureMode
  EnableIntegration
  HostProcesses
  HubFailureMode
  LogLevel
  LogMaxNumFiles
  LogMaxSizeBytes
  OperationMode
  ReprocessClientEmails
  SenderAddressInclusionFilter
  UpdateConfig
- Interactive Warning registry values (see page 29)
  EnableInteractiveWarnings
  LocalMailboxSMTPAddress
  MaxPendingWarnings
  UnmatchedResponseTitle
  UnmatchedResponseTemplateFile
  WarningHeedTimeoutMins
  WarningTemplateFile

- Manually created registry values (see page 31)
  AllowedRecoveryHostProcesses
  CreateEML
  CreateEVF
  DiagnosticFolder
  EnterpriseDNSList
  IntercomPort
  NotificationFromAddress
  ProcessAPMStateValue
  SmtpDNSHostName

More information:

Configure the Exchange, Domino or IIS SMTP Agent (see page 18)

Automatic Registry Values

The following registry values are created automatically when you install the Exchange or Domino server agent or the IIS SMTP agent:

Agent

Available for the Exchange 2003 server agent and IIS SMTP agent only.

Type: REG_SZ

Data: Defaults to Exchange. This registry value is set to either Exchange or IIS. CA DataMinder uses this value to determine the context when processing emails.

If set to IIS, this instructs the IIS SMTP agent to check the DomainMapping registry key. This is only applicable if you are using the IIS SMTP agent to host a CA DataMinder service for use by other organizations.

AttachOriginalEmail

Type: REG_DWORD

Data: Specifies whether the original email is included as an attachment in 'Blocked' notification messages. Set this to zero to omit the original email from these notification messages.

If this value is not present or is set to 1, the email server agent reverts to its default behavior. That is, the original email is attached.

Note: All email attachments are removed from the original message when it is attached to a notification email.

EMailFailureMode

Type: REG_SZ

Data: This value can be set to Delete, Allow, or Mark. It defaults to Allow. This specifies how the email server agent handles event failures. If set to:

- Delete, the server agent deletes the email without notifying the sender.
- Allow, the server agent allows the email to transit through Exchange Server, IIS SMTP, or Domino without intervention.
- Mark, the server agent allows the email to transit through Exchange Server, IIS SMTP, or Domino without intervention, but marks it as if it had been processed normally. This prevents a downstream server agent from re-processing the email.

Email failures can occur when:

- The hub's HighWaterMarkEventCount or HighWaterMarkMB thresholds are exceeded and the HubFailureMode registry value flags all subsequent emails as failures
- The GlobalEventTimeoutSeconds hub timeout expires, or
- There is a general failure when the policy engine analyzes the email.

For details of these hub registry values, see Policy Engine Hub Key (see page 102).

EnableIntegration

Type: REG_DWORD

Data: Defaults to 0 (zero). Determines whether or not the server agent processes emails. To turn on email integration (that is, to enable email processing), set this registry value to 1.

HostProcesses

Available for the Domino server agent only.

Note: We strongly recommend that any changes to this registry value are made only under the guidance of CA technical staff. See CA Technical Support.

Type: REG_SZ

Data: Defaults to nserver.exe,nhttp.exe,nsmtp.exe, nbes.exe,wgnimp.exe,wgnimpsv.exe.

This registry value enables the Domino server agent to process all Domino emails, including those originating from BlackBerry handhelds. In technical terms, this value specifies a list of hooked processes for which the Domino server agent will process Domino callbacks.

If third party email processing applications (such as antivirus or archiving products) are also running on the Domino host machine, you can use this registry value to specify a comma-separated list of process executables associated with these applications instead of the default processes listed above.

The Domino server agent will then use this amended process list to avoid contention with the third party application when processing individual emails. In effect, this registry value (usually in conjunction with the ProcessAPMStateValue registry value) ensures that the Domino server agent only processes the mail after the specified processes have modified an email for the final time.

HubFailureMode

Type: REG_SZ

Data: Defaults to Fail. This value can be set to Wait or Fail. This value instructs the hub what to do if the HighWaterMarkEventCount or HighWaterMarkMB thresholds are exceeded. If set to:

- **Wait**, the hub stops accepting emails from the email server agent until the event queue shortens to reduce allocated memory amount below the relevant 'low water mark' threshold.
- **Fail**, the hub flags all subsequent emails as failures as soon as a 'high water mark' threshold is exceededmail failures are returned to the Exchange, Domino server agent, or IIS SMTP agent, where their handling is controlled by the EMailFailureMode registry value (see above). Normal operations resume when the event queue shortens to reduce allocated memory amount below the relevant 'low water mark' threshold.

LogLevel

Type: REG_DWORD

Data: Defaults to 2. This determines the level of logging for message processing. For example, you can configure the email server agent to only log errors or warning system messages. Log entries are written to the following files:

- Exchange 2003 logs: WgnSmtpS_<date>.log
- Exchange 2007 or 2010 logs: WgnESA_<date>.log
- IIS SMTP logs: WgnSmtpS_<date>.log
- Domino logs: WgnEMNO_<date>.log

Where <date> is the date and time when the log file was created; the file is located in CA's \data\log subfolder of the Windows All Users profile. The supported logging levels are:

1 Errors only

2 Errors and warnings

3 As 2, plus informational and status messages

Note: Setting LogLevel=3 will cause the log file to grow extremely rapidly. This level of logging is provided for testing purposes only.

LogMaxNumFiles

Type: REG_DWORD

Data: Defaults to 10. This specifies the maximum number of log files. When the maximum number of log files exists and the maximum size of the latest is reached (see above), the oldest log file is deleted to enable a new one to be created.

Note: If you are using Exchange Server 2007 or 2010, this registry value is not created automatically.

LogMaxSizeBytes

Type: REG_SZ

Data: Defaults to 1,000,000. This specifies the maximum size for each log file. When the current log file reaches its maximum size, the email server agent creates a new log file.

Note: If you are using Exchange Server 2007 or 2010, this registry value is not created automatically.

OperationMode

Applies to Exchange 2003 and IIS SMTP only.

Important! This setting must not be changed! It currently only supports a value of LocalHub. If this value is changed, you must change it back to LocalHub and restart the IIS service.

Type: REG_SZ

Data: Currently only a value of LocalHub is supported. This specifies that the email server agent passes all events to the policy engine hub on the local machine.

ReprocessClientEmails

Type: REG_SZ

Data: Create and edit this value if your CA DataMinder installation uses both the Outlook client agent and Exchange server agent, or both the Notes client agent and Domino server agent. If this value is present, the default behavior for the server agent corresponds to 'Never'.

Never

Emails marked as already processed by the Outlook or Notes client agent are never processed a second time by the Exchange or Domino server agent.

Quarantine

Emails marked as already processed by the Outlook or Notes client agent can be processed a second time by the Exchange or Domino server agent, but only to apply a quarantine control action. This option is designed to minimize the load on the email server agent.

Always

Emails marked as already processed by the Outlook or Notes client agent are always processed a second time by the Exchange or Domino server agent.

Note: Emails generated by CA DataMinder (for example, automatic replies to blocked emails), are not processed by the Exchange or Domino server agent, even if this registry value is set to 'Always'.

SenderAddressInclusionFilter

Type: REG_SZ

Data: Defaults to *. This value specifies a list of addresses for the email server agent to filter against. Specify a pattern for each version of user's mail (Exchange & SMTP on Exchange Server and Domino & SMTP on a Domino server).

In Active Directory, a user may have many types of email addresses. On an Exchange server the addresses might look as follows:

- Exchange address: /o=Bloggscorp/ou=First Administrative Group/cn=Recipients/cn=fbloggs
- Primary SMTP: fred.bloggs@bloggscorp.com
- SMTP aliases: fred@bloggscorp.com fbloggs@bloggscorp.com fred.bloggs@bloggs.co.uk

On a Domino server, a user might have the following:

- Domino address CN=Fred Bloggs/O=BloggsCorp
- SMTP address fred@bloggscorp.com

For example, if you only want to monitor emails sent from the unipraxis.com domain, you need only set this value to 'unipraxis'. By default, the email server agent monitors *all* emails passing through Exchange Server, IIS SMTP, or Domino.

If the sender's email address does not match any item in this list, the email server agent disregards the email and allows the email to transit through Exchange Server, IIS SMTP, or Domino without intervention.

Similarly, if you want to test that integration with Exchange Server, IIS SMTP, or Domino is working correctly before you go live, you can specify the full address of a test user. This ensures that if there is problem with the integration, no other users will be affected.

Note that * and ? wildcards are supported. A * wildcard matches any sequence of zero or more digits, letters or punctuation characters. For example, a *unipraxis* filter matches spencer@sales.unipraxis.com. A ? wildcard matches any digit, letter or punctuation character, for example, spen?er matches Spencer or Spenser.

UpdateConfig

Available for the Exchange 2007 and 2010 server agents only.

Type: REG_DWORD

Data: Enables administrators to update the registry and the content of any template files while the Exchange server agent is running.

Set to 1 to force the Exchange server agent to reread the registry and the content of any template files. When the Exchange server agent has accepted the changes, it automatically resets this registry value to 0. If the Exchange server agent fails to accept the changes, it automatically sets this registry value to 2.

Important! You must update this registry value in order to update your Exchange server agent configuration.

More information:

Policy Engine Hub Key (see page 102)
Hosting a CA DataMinder Service for Use by Other Organizations (see page 51)

Interactive Warning Registry Values

Available for the Exchange server agent and IIS SMTP agent only.

The following registry values are created automatically when you install the Exchange server agent. They are used to enable and configure server-side interactive warnings.

EnableInteractiveWarnings

Type: REG_DWORD

Data: Default to (0) zero. Set this value to 1 to enable interactive warning messages.

LocalMailboxSMTPAddress

Type: REG_SZ

Data: Defaults to empty. Set this value to the SMTP address of the Compliance Release mailbox. This is the Reply To address for interactive warning notifications.

MaxPendingWarnings

Type: REG_WORD

Data: Defaults to 5000. This value specifies the number of ‘pending warnings’ that can be held on the Exchange server. That is, the number of emails awaiting a response from a previously sent warning notification message. When the maximum number of pending warnings is reached, the oldest warning is automatically heeded to make room for a new ‘pending warning'.

UnmatchedResponseTitle

Type: REG_SZ

Data: Defaults to %subject%. This value specifies the title text for the unmatched response message. It defaults to an insertion variable corresponding to the subject of the user's reply to the warning message.

Note: For details on supported insertion variables, see the Administration console online help; search for 'interactive warnings'.

UnmatchedResponseTemplateFile

Type: REG_SZ

Data: This value references a text file containing the body text for the unmatched response message. Enter the full path to the relevant file. Defaults to:
`\ProgramFiles\CA\CA DataMinder\Client\UnmatchedResponseTemplate.txt`

Note: This text file can be in ANSI, UTF-8, Unicode, or HTML format. It can also contain insertion variables that refer to the user's reply to the warning message. For details on supported insertion variables, see the Administration console online help; search for 'interactive warnings'.

WarningHeedTimeoutMins

Type: REG_DWORD

Data: Defaults to 240 (four hours). This value specifies how long (in minutes) a pending warning remains on the Exchange server. After this timeout expires, the warning is automatically heeded.

WarningTemplateFile

Type: REG_SZ

Data: This value references a text file containing the body text for the warning message. Enter the full path to the relevant file. Defaults to:

`\ProgramFiles\CA\CA DataMinder\Client\WarningTemplate.txt`

Note: This text file can be in ANSI, UTF-8, Unicode, or HTML format and can contain insertion variables.

More information:

Server-side Interactive Warnings (see page 34)

Manually Created Registry Values

The following registry values are also supported. Most, but not all, are for diagnostic purposes. You may need to create these values, for example, to test the integration with Exchange Server before going live:

AllowedRecoveryHostProcesses

Available for the Domino server agent only.

Type: REG_SZ

Data: Specifies a list of processes that can host the recovery thread. This list must be a subset of the processes listed in the HostProcesses registry value.

You can use this registry value to prevent a third-party application process from hosting the recovery thread.

Note: If AllowedRecoveryHostProcesses is not present then the entire list in HostProcesses is used instead. There is only one instance of the recovery thread. That is, only the first process to start the recovery thread will actually run it.

CreateEML

Available for the Exchange server and IIS SMTP agents only.

Type: REG_DWORD

Data: Specifies whether to create diagnostic files containing the emails and associated data for emails transiting through the server. These files are:

EML

Contains the 'raw' MIME email data for each email processed.

EMP

Contains additional data relating to the email, such as the actual recipient list and internal properties.

If this value is set to:

0, the server never creates EML/EMP files.

1, the server always creates EML/EMP files and saves them in the DiagnosticFolder.

2, the server creates EML/EMP files on error and saves them in the DiagnosticFolder.

Important! We strongly recommend that you do not change this setting unless directed to do so by CA Support.

CreateEVF

Type: REG_DWORD

Data: Specifies whether an EVF file is created for each email in CA DataMinder EVF format processed by the policy engine hub. If this value is set to:

0, the server never creates EVF files.

1, the server always creates EVF files and saves them in the DiagnosticFolder.

2, the server creates EVF files on error and saves them in the DiagnosticFolder. That is, when an event returns from the policy engine hub with an error. This value is only available for the Exchange 2007 and 2010 server agents.

DiagnosticFolder

Type: REG_SZ

Data: Specifies the path and folder where diagnostic files are saved. The creation of these files is determined by the CreateEML and CreateEVF registry values.

Note: This folder is *not* created automatically.

EnterpriseDNSList

Available for Exchange and Domino server agents and IIS SMTP agents.

Type: REG_SZ

Data: Defaults to empty, but see 'Updating an empty list' below. This registry value specifies a comma-separated list of DNS domains that you want to be considered as a single enterprise. It is typically used with SmtpDNSHostName (see below). Its purpose is to simplify the method for verifying that emails are not reprocessed needlessly by consecutive email server agents.

Note: Values in this list can only contain ASCII characters, without spaces or control characters (for example, tab spaces). Use commas as a list separator.

How does this work? The local email server agent assumes each of the domains listed in EnterpriseDNSList has its own email server agent, and that any email arriving from a listed domain has already been processed by CA DataMinder and does not need reprocessing.

In technical terms, when a remote email server agent processes an email, it writes the DNS domain of the Exchange, Domino or SMTP server to the email's MIME tag. Or, if the SmtpDNSHostName registry value has been configured, this value is written to the MIME tag instead.

When the *local* email server agent receives this email, it checks the MIME tag. If the source domain matches a domain in EnterpriseDNSList, the server agent does not reprocess the email.

Updating an empty list: While this list is empty, the DNS domain of the local Exchange, Domino or SMTP server is implied. That is, the local email server agent does not reprocess emails arriving from this domain.

IntercomPort

Available for the Exchange 2007 and 2010 server agents and the IIS SMTP agent.

Type: REG_DWORD

Data: Specifies the inter-agent communications port number. Port numbers between 1 and 65535 are valid in this setting. Defaults to 8102.

Note: If this parameter is changed, you must then restart the Exchange server agent or IIS SMTP agent.

NotificationFromAddress

Type: REG_SZ

Data: Specifies a text string representing the 'notification sender'. This text appears in the From: box in all notification emails generated by Exchange or Domino server agents (such as when an email is blocked). For example, you can specify that the From: field in notification emails is always set to ComplianceTeam@unipraxis.com.

This registry value is optional and overrides the default sender (typically 'postmaster'). The text string must be a valid SMTP address.

ProcessAPMStateValue

Available for the Domino server agent only.

Important! We strongly recommend that any changes to this registry value are made only under the guidance of CA technical staff. For details, contact CA Support at http://ca.com/support.

Type: REG_SZ

Data: This optional registry value specifies a value for the 'APMState' item that the Domino server agent waits for before it processes a note.

You can use this registry value to enable the Domino server agent to wait for a third-party application to finish processing, before it applies policy. However, this is normally only used with the HostProcesses registry value.

Note: If this registry value is used, it must be set to the text string that the third-party application sets as the value of the APMState item. For example, 3rdPartyComplete.

SmtpDNSHostName

Available for Exchange and Domino server agents and IIS SMTP agents.

Type: REG_SZ

Data: Defaults to empty. This value specifies a valid DNS name that complies with RFC naming conventions. It is always used with EnterpriseDNSList (see above). Its purpose is to simplify the method for verifying that emails are not reprocessed needlessly by consecutive email server agents.

How does this work? SmtpDNSHostName specifies a single DNS domain that is written to the email's MIME tag after it has been processed by the local email server agent. If set, this registry value overrides the DNS domain of the Exchange, Domino or SMTP server in the MIME tag.

To use this registry value as intended, you need to set SmtpDNSHostName to the same value (for example, UNIPRAXIS.COM) for all your email server agents. You can optionally include this value in the EnterpriseDNSList domain list. Now, when any email server agent receives an email tagged as coming from UNIPRAXIS.COM, it knows that policy has already been applied to the email and so does not reprocess it.

Server-side Interactive Warnings

Note: Interactive warnings are only available for the Exchange server agent and the IIS SMTP agent.

If CA DataMinder intercepts an email transiting through Exchange Server or IIS SMTP and the email generates a warning or inform event, CA DataMinder can automatically send a notification or an interactive warning email to the sender.

If the sender replies to this warning promptly (that is, before the warning timeout expires), then their email is released and sent to its intended recipients. If they do not reply (or reply too late), then CA DataMinder deems that they have heeded the warning and the email is disposed of without being released. The warning timeout defaults to 4 hours. That is, a user has 4 hours to reply if they want to disregard the warning and send their email anyway. But this timeout is configurable.

Interactive Warnings and Followup Messages

The sender of an email can receive a warning and, if necessary, a follow-up message from the email server agent. The text content of these messages is configurable, by using template text files referenced in the registry

Warning message

This is the first message the user receives. It is sent automatically when the agent intercepts an email that has generated a warning or inform event. By default, the message has the user's original email attached and states that it has triggered one or more warnings. It lists the warnings and advises the sender that if they want the email to be sent, they must reply to the warning message.

'Unmatched response message'

This message is automatically sent when the user replies to the warning message, but the original email is no longer on the Exchange or IIS server. In this situation, the user's reply cannot be matched to the original email and so the email cannot be released and sent. Replies are matched to their corresponding emails by a unique ID in the Subject. The original email may no longer be on the Exchange or IIS server for any of the following reasons:

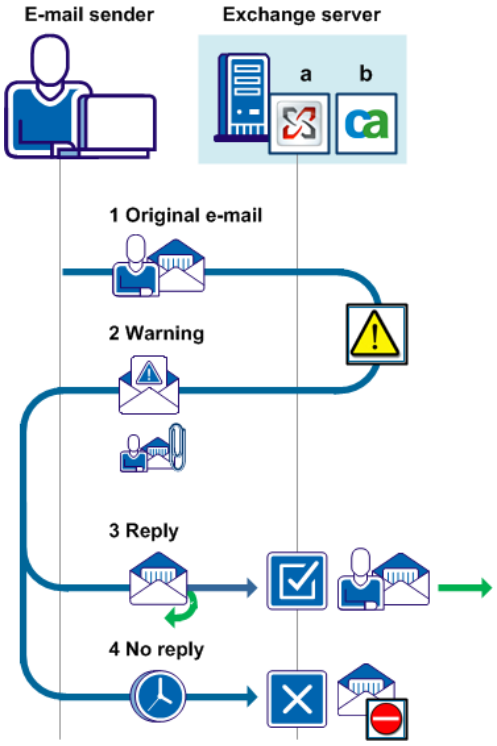
- The user replied too late and the warning 'autoheed' timeout expired. (This timeout is defined by WarningHeedTimeoutMins.)
- The user replied to the warning more than once. The first reply matches the original email and allows it to be sent, which in turn, removes it from the Exchange or IIS server. Any subsequent replies therefore cannot be matched.
- The maximum number of pending warnings was reached and the user was unable to reply to the warning message before it was autoheeded.
- (Exchange 2007 and 2010 server agents, IIS SMTP agent) The agent holding the pending email and the agent holding the response email (the reply from the user) cannot communicate.

More information:

Interactive Warning Registry Values (see page 29)

Exchange Server 2003

If using Exchange Server 2003, the CA DataMinder Exchange server agent must exist on the Exchange server to enable integration for interactive server-side warnings. The server-side interactive warnings procedure is summarized below:



1. A user sends an email. As it transits through Exchange (**a**), it is detected by the Exchange server agent (**b**) and triggers a warning.

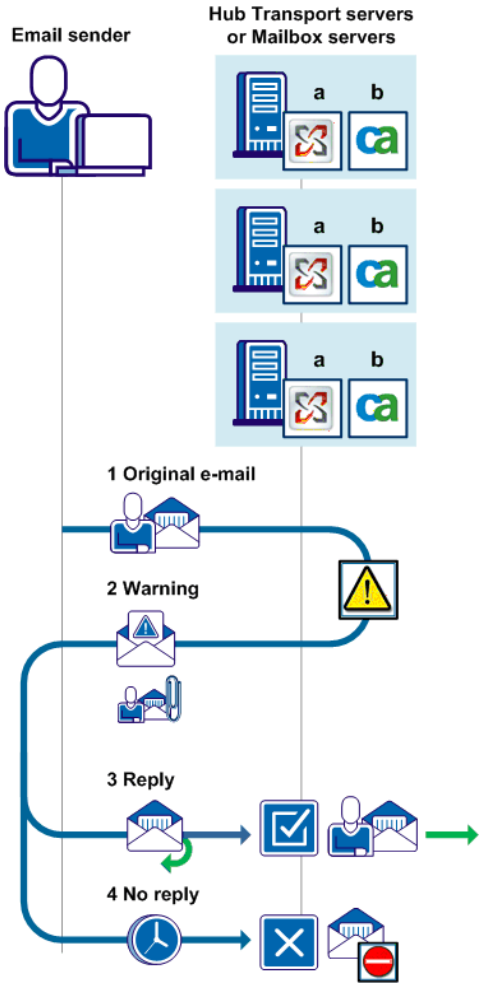
2. The Exchange server agent sends a warning email, with the offending original email included as an attachment.

3. If the sender replies to the warning, their email is released and sent on to its intended recipients. The event is saved on the CMS as a 'disregarded warning'.

4. If the sender does not reply before the warning timeout expires, the e-mail is not sent. The event is saved on the CMS as a 'heeded warning'.

Exchange Server 2007, 2010, and 2013

Exchange Server 2007, 2010, and 2013 are modular systems, so your Exchange 'site' may include multiple Hub Transport servers or Mailbox servers. To enable server-side interactive warnings for these versions of Exchange, the Exchange server agent must exist on each Hub Transport server (Exchange 2007 or 2010) or each Mailbox server (Exchange 2013). The server-side interactive warnings procedure is summarized below:



1. A user sends an email. As it transits through Exchange (**a**), it is detected by the Exchange server agent (**b**) on one of the Hub Transport servers (Exchange 2007 or 2010) or Mailbox servers (Exchange 2013) and triggers a warning.

2. The Exchange server agent sends a warning email, with the offending original email included as an attachment.

3. If the sender replies to the warning, their email is released and sent on to its intended recipients. The event is saved on the CMS as a 'disregarded warning'.

4. If the sender does not reply before the warning timeout expires, the email is not sent. The event is saved on the CMS as a 'heeded warning'.

IIS SMTP

If using IIS SMTP, the CA DataMinder IIS SMTP agent must exist on the IIS server to enable integration for interactive server-side warnings. If multiple IIS servers are deployed for load-balancing purposes to serve the same SMTP routes, then the CA DataMinder IIS SMTP agent must exist on each of the IIS servers.

The server-side interactive warnings procedure is similar to the one described for Exchange 2007 and 2010, reading "IIS server" whenever "Hub Transport Server" is mentioned.

Deployment Procedure for Interactive Warnings

To enable interactive warning messages, you create a 'Compliance Release' mailbox. The Exchange server agentor IIS SMTP agent use this mailbox to send warnings and receive replies to these warnings. The procedure is summarized below.

To set up email server agents to send interactive warning emails

1. Create your Compliance Release mailbox.

   For Exchange, create a 'Compliance Release' mailbox on each Exchange server (or site) where you want interactive warning emails to work.

   For IIS SMTP servers, do one of the following:

   - Configure SMTP routing so that emails sent to the 'Compliance Release' mailbox address are routed to the IIS server hosting the IIS SMTP agent. (You do not technically need a mailbox with this solution.)
   - Install a Mail Server role on the IIS server, and create a mailbox to serve as the 'Compliance Release' mailbox.

2. Configure the agent to use the Compliance Release mailbox.

   To do this, edit the LocalMailboxSMTPAddress registry value on the host server(s).

3. Define your message templates.

   The templates include the text for the warnings and 'unmatched response' messages seen by the sender.

   The template files are installed with the Exchange server agent or IIS SMTP agent.

4. Enable interactive warnings.

   To do this, edit the EnableInteractiveWarnings registry value on the host server(s).

These steps are described in the following sections.

Create a Compliance Release Mailbox for Exchange Agents

The procedure setting up interactive warnings varies according to which version of Exchange you are using. The steps are summarized below:

1. Create a Compliance Release mailbox:

   Exchange Server 2003

   Create a Compliance Release mailbox on every Exchange server you want to configure for interactive warnings.

   Exchange Server 2007 or 2010

   To set up interactive warnings on your Hub Transport servers, create a Compliance Release mailbox account for each Active Directory site. Then associate this account with a mailbox hosted by that site. You only need to create one mailbox account for each site.

   Exchange Server 2013

   Choose a deployment method:

   - **1:1 Mailboxes:** Create a Compliance Release mailbox account on each Mailbox server. Each Exchange server agent has its own Compliance Release mailbox. This method ensures that interactive warnings continue to operate if a Mailbox server is unavailable.
   - **Shared Mailbox:** Create a single Compliance Release mailbox on one Mailbox server. All Exchange server agents share the single Compliance Release mailbox. This method is similar to how previous versions of CA DataMinder integrated with Exchange 2007 and 2010 (where a single mailbox for each Active Directory site served multiple Hub Transport servers).

2. Set the primary SMTP address of the mailbox.

   To enable CA DataMinder to identify emails destined for the mailbox you created in step 1, this mailbox must have 'cadlp.' as the first element of its primary SMTP address. For example:

   Exchange Server 2003

   ```
   cadlp.<servername>@<domain>
   ```

   Exchange Server 2007 or 2010

   ```
   cadlp.<sitename>@<domain>
   ```

   Exchange Server 2013 with 1:1 mailboxes

   ```
   cadlp.<servername>@<domain>
   ```

   Exchange Server 2013 **with a shared mailbox**

   ```
   cadlp.<sitename>@<domain>
   ```

3. Create a new mailbox-enabled user in Exchange.

   We recommend that you specify an Exchange mailbox alias when you create the mailbox and that this alias starts with:

   Exchange Server 2003
   `cadlp.<servername>`

   Exchange Server 2007 or 2010
   `cadlp.<sitename>`

   Exchange Server 2013 with 1:1 mailboxes
   `cadlp.<servername>`

   Shared Mailbox with a shared mailbox
   `cadlp.<sitename>`

   We also recommend that you change the mailbox Display Name or Full Name to 'Compliance Release'. This name displays in the To: field in the user's reply to a warning message. (You can set the name when you create the user, or you can change it afterwards.)

4. (Optional) Manually add an SMTP address to the mailbox user that conforms to the alias recommendations in step 3.

   You only need to do this if:

   - You did *not* follow the mailbox alias recommendations in step 3, and
   - Exchange Server does *not* have a recipient policy that automatically creates an SMTP address based on the Exchange mailbox alias (that is, with %m in the userformat).

   If you do need to manually add an SMTP address:

   - Use the correct address format (see step 2).
   - Designate the address as the primary SMTP address for the mailbox. In Exchange 2003, use the 'Set As Primary' option. In Exchange 2007 and 2010, use the 'Set as Reply' option. For Exchange 2013, you choose the 'Make this the reply address' check box. For full details, see your Exchange Server documentation.

Create a Compliance Release Mailbox for IIS SMTP Agents

IIS SMTP agents differ from Exchange server agents in that there is not normally a mail server associated with the SMTP service, unlike in Exchange Servers, which are available to create a 'Compliance Release' mailbox on.

Interactive warnings work as follows with the IIS SMTP agent:

- If an email sent by a user triggers a warning, the agent that intercepted the email sends the user a warning message.
- To release the original email, the reply to that warning must arrive back at the agent that is holding the original email pending.

  With the Exchange server agent, that is achieved by creating a mailbox (the 'Compliance Release' mailbox) on the Exchange Server. Replying to the warning message automatically sends the reply to the 'Compliance Release' mailbox. The holding agent is guaranteed to receive the reply because the mailbox is hosted by the Exchange Server which the agent serves (with Exchange 2007 or 2010, or by multiple agents in the case of Exchange 2007 or 2010, one of which will be the holding agent).

  With the IIS SMTP agent, there are two different ways to make sure that the reply arrives at the holding agent:

  - Configure SMTP routing so that emails sent to the 'Compliance Release' mailbox address from the managed users are routed to the IIS server hosting the agent (or to one of the IIS servers hosting agents, in the case of multiple IIS servers serving the same SMTP routes). You do not technically need a mailbox with that solution.
  - Install a Mail Server role on the IIS server, and create a mailbox to serve as the 'Compliance Release' mailbox. Make sure to configure the Mail Server for the correct SMTP routing.

    Note: If multiple IIS servers are deployed for load-balancing purposes to serve the same SMTP route(s), only install a Mail Server role on one of the IIS servers.

  To enable CA DataMinder to identify emails destined for the 'Compliance Release' mailbox, this mailbox (or address, if there is no actual mailbox) must have 'cadlp.' as the first element of its SMTP address. For example:

  ```
  cadlp.<IISServer>@<YourDomain>
  ```

Specify the Compliance Release Mailbox

The Exchange server agent or IIS SMTP agent needs to know which mailbox to use for interactive warning emails and replies to these warnings. Specifically, you must configure the agent to use the Compliance Release mailbox.

To specify the Compliance Reliance mailbox

1. Locate the following registry key.
   `HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder\CurrentVersion\Exchange`

   See below for details about where to find this registry key.

2. Set the LocalMailboxSMTPAddress registry value to the SMTP address of the Compliance Release mailbox that you previously created.

   Exchange Server 2007 and 2010

   The 'Local' element in LocalMailboxSMTPAddress is now a misnomer. This registry value was named when CA DataMinder first supported Exchange 2003. When later CA DataMinder releases added support for Exchange 2007 and 2010, this registry value no longer referred to a local mailbox. However, we have not renamed LocalMailboxSMTPAddress in order to retain compatibility with older versions of CA DataMinder when customers upgrade.

   Exchange Server 2013

   If you chose the '1:1 Mailboxes' deployment method, set LocalMailboxSMTPAddress to the SMTP address of the *local* Compliance Release mailbox used by the *local* Exchange server agent.

   If you chose the 'Shared Mailbox' deployment method, set LocalMailboxSMTPAddress to the SMTP address of the *shared* Compliance Release mailbox used by all Exchange server agents.

3. (Exchange 2007, 2010, or 2013 only) Set the UpdateConfig registry value to 1.

Where do I find the \Exchange registry key and these registry values?

For Exchange Server 2003, find this registry key and these registry values on your Exchange server.

For Exchange Server 2007 and 2010, find this registry key and these registry values on each Hub Transport server.

For Exchange Server 2013, find this registry key and these registry values on each Mailbox server.

For IIS SMTP servers, find this registry key and these registry values on each IIS server that hosts an IIS SMTP agent. (If multiple IIS servers are deployed for load-balancing purposes to serve the same SMTP routes, edit the registry values on each IIS server.)

More information

Email Server Agent Registry Values (see page 22)
Interactive Warning Registry Values (see page 29)

Define the Message Templates

You must set up the templates for the warning message and 'unmatched response' follow-up message.

To define the message templates

1. Define the body text for warning messages and 'unmatched response' messages.

   You specify the body text for these messages in two separate text files, WarningTemplate.txt and UnmatchedResponseTemplate.txt. These files are installed with the Exchange server agent or IIS SMTP agent in the following folder on the host server:
   `\Program Files\CA\CA DataMinder\Client`

2. (Applicable only if you use non-default paths or file names) Specify the path and file name of these text files by editing these registry values:

   WarningTemplateFile

   UnmatchedResponseTemplateFile

3. (Optional) Specify the subject field for unmatched response' messages by editing the UnmatchedResponseTitle registry value.

4. (Exchange 2007, 2010, or 2013 only) Set the UpdateConfig registry value to 1.

Where do I find these files and registry values?

For Exchange Server 2003, find these files and registry values on your Exchange server.

For Exchange Server 2007 and 2010, find these files and registry values on each Hub Transport server.

For Exchange Server 2013, find these files and registry values on each Mailbox server.

For IIS SMTP servers, find these files and registry values on each IIS server that hosts an IIS SMTP agent.

More information

Interactive Warning Registry Values (see page 29)
Message Templates Files (see page 45)

Enable Interactive Warnings

Finally, turn on support for interactive warnings on the Exchange server or IIS server.

To allow the email server agent to send interactive warning emails

1. Edit the EnableInteractiveWarnings registry value.
2. (Optional) You can also modify the default timeout for heeded warnings.

   Edit the WarningHeedTimeoutMins registry value.
3. (Exchange 2007, 2010, or 2013 only) Set the UpdateConfig registry value to 1.

Where do I find these registry values?

For Exchange Server 2003, find these registry values on your Exchange server.

For Exchange Server 2007 and 2010, find these registry values on each Hub Transport server.

For Exchange Server 2013, find these registry values on each Mailbox server.

For IIS SMTP servers, find these registry values on each IIS server that hosts an IIS SMTP agent.

More information:

Email Server Agent Registry Values (see page 22)

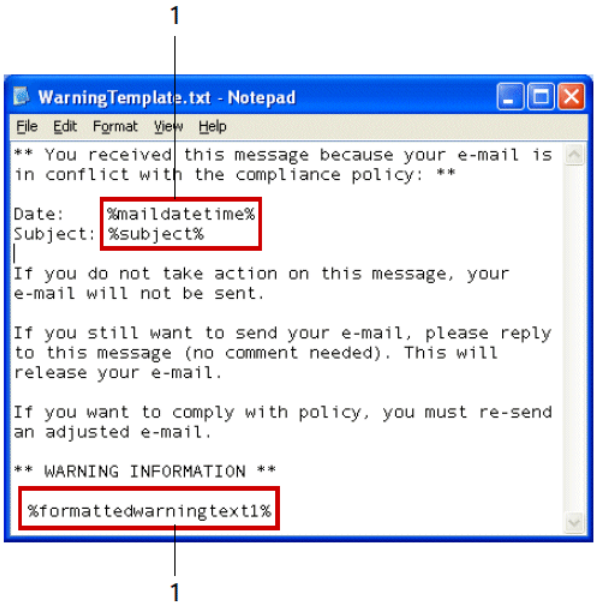
Message Templates Files

The template text files for the warning and 'unmatched response' messages can be in ANSI, UTF-8, Unicode, or HTML format and can also include insertion variables.

Default template files, WarningTemplate.txt and UnmatchedResponseTemplate.txt, are installed with the Exchange or IIS server agent into the following folder on the machine hosting the server agent:

\Program Files\CA\CA DataMinder\Client

The default warning message template is shown below. To overwrite or amend the default message templates, you need to edit the registry on your Exchange or IIS server.



Default template for warning messages

1 Insertion text variables.

Template Insertion Variables

CA DataMinder supports the following variables in notification message template files:

%subject% variable

When used in a warning message template, this variable is replaced by the subject of the original email in the message seen by users.

When used in an 'Unmatched Response' message template, this variable is replaced by the subject of the user's reply to the warning message.

This variable can also be typed directly into the UnmatchedResponseTitle registry value on the Exchange server, where this registry value sets the 'unmatched response' message title.

%maildatetime% variable

When used in warning message templates, this variable is replaced by the date and time of the original email, relative to the time zone of the host server.

When used in 'Unmatched response' message templates, this variable is replaced by the date and time of the user's reply to the warning message, relative to the time zone of the host server.

The date and time in the message seen by users is displayed in RFC 2822 format. For example, if the template file contains this text:

```
Your email on %maildatetime% appears to breach corporate guidelines.
```

It displays like this in the warning:

```
Your email on Fri, 12 May 2006 16:22:10 -0400 appears to breach corporate
guidelines.
```

Where, in this example, -0400 indicates EDT (Eastern Daylight Time) or four hours behind UTC.

%formattedwarningtext1% variable

When used in warning message templates, this variable is replaced by details of the warnings (or inform events) triggered by the email. For each trigger that activates, this variable returns two text items: a title and message:

- The title derives from the 'Dialog Title - Warnings and Quarantine Events' setting in the System Settings\User Notifications\ policy folder (or for Inform events, the 'Dialog Title - Inform, Notify, and Categorize Events' setting).
- The message is based on the Message To Users setting, included in each policy trigger.

If the sender's email causes multiple triggers to activate, the %formattedwarningtext1% variable writes to the warning email a 'title and message' pair for each trigger, with each pair separated by a blank line:

```
<Warning dialog title 1>
<Trigger 1 message>
<Blank line>
<Warning dialog title 2>
<Trigger 2 message>
```

For example, if two triggers activate, the warning email would look like this:

```
CA DataMinder Advisory
You are not permitted to send emails to these teams: Equity Markets, Debt Markets.

CA DataMinder Advisory
This email refers to 'Project Alpha'. Such references are normally prohibited in
corporate correspondence.
```

Note: This variable is not suitable for use with the UnmatchedResponseTemplateFile registry value or the UnmatchedResponseTitle message template.

%formattedwarningtext2% variable

When used in warning message templates, this variable is replaced by the details of the warnings (or inform events) triggered by the email. For each trigger that activates, this variable returns a message based on the 'Message To Users' trigger setting.

If the sender's email causes multiple triggers to activate, %formattedwarningtext2% writes to the warning email a message for each trigger, separated by a blank line:

```
<warning #1 message>
<blank line>
<warning #2 message>
```

Note: This variable is not suitable for use with the UnmatchedResponseTemplateFile registry value or the UnmatchedResponseTitle message template.

%to% variable

When used in warning message templates, this variable is replaced by the address(es) in the To field of the original email. For example, if the template file contains:

```
Your email to %to% breaches corporate rules.
```

It displays like this in the warning:

```
Your email to srimmel@unipraxis.com breaches corporate rules.
```

Note: This variable is not suitable for use with the UnmatchedResponseTemplateFile registry value or the UnmatchedResponseTitle message template.

%cc% variable

When used in warning message templates, this variable is replaced by the address(es) in the Cc field of the original email.

Note: This variable is not suitable for use with the UnmatchedResponseTemplateFile registry value or the UnmatchedResponseTitle message template.

Maximum Number of Pending Warnings

To prevent a backlog of emails awaiting a warning reply from accumulating, CA DataMinder sets a maximum limit on the number of warnings pending.

To configure this limit, you need to edit the MaxPendingWarnings registry value on the machine hosting the Exchange server agent or IIS SMTP agent.

When the maximum number of pending warnings is reached the oldest warning is automatically heeded, enabling another pending warning to be held.

'Autoheed' Warning Timeout

For each warning message, CA DataMinder sets a timeout. If no reply to the warning is received before the timeout expires, the warning is deemed to have been heeded by the sender. That is, the email is not delivered to its intended recipients and is saved as a 'heeded warning' event on the CMS. This timeout defaults to 4 hours. We recommend you keep it reasonably short to reduce the risk of policy changing during this time.

To configure this timeout, you need to edit the WarningHeedTimeoutMins registry value on the machine hosting the Exchange server agent or IIS SMTP agent.

Policy Checks

Emails that generate warnings are subject to two policy checks. The first occurs when the email is originally sent: this policy check triggers the warning message. The second occurs immediately before the Exchange server agent or IIS SMTP agent resumes processing the email (either because the user has replied to the warning message, or because the Auto Heed warning timeout has expired).

Normally, the policy engine detects that policy has *not* changed between the two checks. If the user:

- **Replied** to the warning message (that is, the user disregarded the warning), the original email is sent.
- **Did not reply** to the warning message (that is, the user heeded the warning), the original email is deleted unsent.

Sometimes however, the policy engine may detect that policy *has* changed. If this is the case and the user:

- **Replied** to the warning message (that is, the user disregarded the warning), the original email is checked against the new policy. If the email violates the new policy, the sender receives a new interactive warning message. If the email does not violate the new policy, it is sent.
- **Did not reply** to the warning message (that is, the user heeded the warning), the original email is deleted unsent.

Note: The 'autoheed warning' timeout can expire prematurely if the maximum number of pending warnings is reached.

More information:

'Autoheed' Warning Timeout (see page 48)

Turn on Exchange, Domino or IIS SMTP Integration

Important! You must enable CA DataMinder email server agents in order to start email processing. By default, the agents are disabled when first deployed. If you deployed multiple agents, enable each agent individually!

The Exchange and Domino server agents, and the IIS SMTP agent use a single registry value to turn on email integration with Exchange Server, IIS SMTP, or Domino. Because any changes to this value are effective immediately, we recommend that you make this the final step when setting up integration with Exchange Server, IIS SMTP, or Domino.

To turn on email integration

1. For Exchange and IIS SMTP integration, locate registry this key:
   `HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder\CurrentVersion\Exchange`

   For Domino integration, locate this key:
   `HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder\CurrentVersion\Domino`

   See below for details about where to find these registry keys.
2. Set the EnableIntegration registry value to 1 to turn on email integration.
3. (Exchange 2007, 2010, or 2013 only) Set the UpdateConfig registry value to 1.

   Email processing is enabled. CA DataMinder monitors all emails transiting through the email server.

To turn off email integration

1. Go to the specified registry key (see above).
2. Set EnableIntegration to 0.
3. (Exchange 2007, 2010, or 2013 only) Set the UpdateConfig registry value to 1.

   Email processing is disabled. CA DataMinder allows all emails to transit through the email server without intervention.

Where do I find these registry keys and registry values?

For Exchange Server 2003, find this registry key and these registry values on your Exchange server.

For Exchange Server 2007 and 2010, find this registry key and these registry values on each Hub Transport server.

For Exchange Server 2013, find this registry key and these registry values on each Mailbox server.

For IIS SMTP servers, find this registry key and these registry values on each IIS server that hosts an IIS SMTP agent. (If multiple IIS servers are deployed for load-balancing purposes to serve the same SMTP routes, edit the registry values on each IIS server.)

For Domino servers, find this registry key and the EnableIntegration registry value on your Domino servers.

More information:

Automatic Registry Values (see page 23)

Set Up SecureMail Integration

You can configure CA DataMinder to detect, analyze, and apply policy to emails encrypted by Voltage SecureMail. Specifically, the following agents can detect and apply policy to emails encrypted by Voltwage SecureMail:

- Exchange server agent
- IIS SMTP server agent
- Milter MTA agent (for Sendmail and Postfix email servers)

To set up SecureMail integration, you must edit the registry on your policy engines and set a shared secret. Policy engines use the shared secret to establish secure connections to the Voltage SecureMail server. Details are in the Voltage SecureMail Integration chapter.

More information:

Voltage SecureMail Integration (see page 79)

Hosting a CA DataMinder Service for Use by Other Organizations

Using the IIS SMTP agent, you can provide a hosted CA DataMinder solution for use by other organizations. In this situation, the IIS SMTP is configured to identify which organization an e-mail sender belongs to. It passes this information with the email to a policy engine to ensure that policy specific to that organization gets applied to the email.

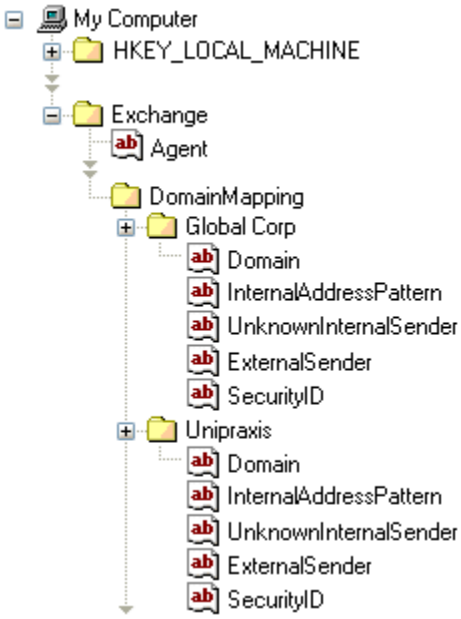
To set this up, you must edit the registry on the machine hosting the IIS SMTP agent. Specifically, you need to create a DomainMapping registry subkey plus, within this subkey, a further subkey for each organization using the hosted CA DataMinder solution. The path to each organization subkey is as follows:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder\CurrentVersion
    \Exchange\DomainMapping\<Organization>
```

These organization subkeys each contain a registry value that associates specific domains with the organization, plus registry values to determine which user policies get applied to emails from unrecognized senders.

Registry Example

This example shows two organization subkeys, for Global Corp and Unipraxis, each containing its own domain mapping registry values plus values for handling emails from unrecognized senders.



Agent registry value

Before creating the DomainMapping subkey, you need to edit the Agent (see page 23) registry value.

DomainMapping registry subkey

Manually add a DomainMapping subkey below the Exchange registry key. This subkey will contain the <Organization> subkeys for each organization using the hosted CA DataMinder service.

***<Organization>* subkeys**

Within the DomainMapping subkey, you need to create a separate subkey for each organization using the hosted CA DataMinder service. Give each a subkey a name that reflects the organization. Next, within each <organization> subkey, you need to create the following registry values:

- Domains
- InternalAddressPattern
- UnknownInternalSender
- ExternalSender
- SecurityID

These values are described below.

Domains registry value

Type: REG_MULTI_SZ

Data: This registry value specifies a list of domain names associated with the current organization.

When the IIS SMTP agent detects a sender address from a listed domain, it associates that email with the relevant organization. When it passes the email to a policy engine for processing, it also passes that organization's message handling details, as specified by the registry values below.

InternalAddressPattern registry value

Type: REG_SZ

Data: This registry value specifies a full or partial email address.

If this value is set, a policy engine checks the sender's email address against this address pattern when it first processes an email. If the sender's address does match this pattern, the policy engine attempts to map the sender onto an existing CA DataMinder user account.

If set, this registry value overrides the corresponding Internal Email Address Pattern setting in the policy engine's machine policy. If this registry value is not set, policy engines revert to the Internal Email Address Pattern policy setting to determine which policy apply.

SecurityID registry value

Type: REG_SZ

Data: Specifies a security ID that is stored with all events associated with the current organization. This security ID is used to segregate events on the CMS, ensuring that reviewers can only retrieve events for their organization.

UnknownInternalSender registry value

Type: REG_SZ

Data: This registry value specifies the name of a CA DataMinder user.

If this value is set, policy engines apply this user's policy to emails sent from someone **within** the organization. The policy engine applies the UnknownInternalSender policy if the sender's address matches an address pattern listed in InternalAddressPattern (see above) but no corresponding user exists.

If set, this registry value overrides the corresponding Unknown Internal Sender setting in the policy engine's machine policy. If this registry value is not set, policy engines revert to the Unknown Internal Sender policy setting to determine which policy apply.

ExternalSender registry value

Type: REG_SZ

Data: This registry value specifies the name of a CA DataMinder user.

If this value is set, policy engines apply this user's policy to external emails. That is, emails sent from someone **outside** the organization. The policy engine applies the ExternalSender policy if the sender's address does **not** match an address pattern listed in InternalAddressPattern (see above).

If set, this registry value overrides the corresponding External Sender setting in the policy engine's machine policy. If this registry value is not set, policy engines revert to the External Sender policy setting to determine which policy apply.

Monitor the Exchange, Domino or IIS SMTP Agents

There are various sources of diagnostic information when monitoring the email server agent, including log files and diagnostic files. Performance counters are also available for policy engines and hubs.

Log Files for Email Server Agents

The email server agents write log entries to the following log files:

- **Exchange 2003:** WgnSmtpS_<date>.log

IIS SMTP: WgnSmtpS_<date>.log

- **Exchange 2007 and 2010:** WgnESA_<date>.log
- **Domino:** WgnEMNO_<date>.log

These files are located in CA's \data\log subfolder of the Windows All Users profile.

You set the log level by editing the LogLevel registry value.

More information:

Automatic Registry Values (see page 23)

Diagnostic Files for Email Server Agents

The email server agents can be configured to generate 'diagnostic' files when extracting the contents of an email. You can specify that diagnostic files are always created, only created if an error occurs, or never created at all. For details, see the following registry values:

- DiagnosticFolder
- CreateEML
- CreateEVF

More information:

Manually Created Registry Values (see page 31)

Uninstall Exchange, Domino or IIS SMTP Agents

Important! If a policy engine is installed on the same computer as an Exchange, IIS, Domino, or Enterprise Vault server agent, you must uninstall the server agent before uninstalling the policy engine.

When you uninstall an email server agent, the hub is also uninstalled automatically. Use Add/Remove Programs to manually uninstall the Exchange or Domino server agents or IIS SMTP agent. This applet is part of the Control Panel.

To uninstall an email server agent

1. In Add/Remove Programs, select CA DataMinder Integration Agents and click Change.
2. When the wizard starts, go to the Program Maintenance screen and choose Modify.

   Note: If you choose Remove, this removes all CA DataMinder components, not just the Exchange or Domino server agents or IIS SMTP agent.
3. In the Custom Setup screen, choose the Exchange Server Agent or Domino Server Agent, or IIS SMTP Agent as required.
4. In the final wizard screen, click Install to being the uninstallation. But see the sections below.

Exchange 2003 Server Agent or IIS SMTP Agent and IIS

When uninstalling the Exchange server agent or IIS SMTP agent, the wizard stops Internet Information Services (IIS) before uninstalling the Exchange server agent or IIS SMTP agent and hub components. It then restarts IIS automatically when the uninstall is complete.

Note: IIS is stopped and started automatically as part of an Exchange server agent ot IIS SMTP agent installation.

Exchange 2007 or 2010 Server Agent and the Microsoft Exchange Transport Service

When uninstalling the Exchange server agent, the wizard stops the Microsoft Exchange Transport Service before uninstalling the Exchange server agent and hub components. It then restarts this service automatically when the uninstall is complete.

Note: The Microsoft Exchange Transport Service is stopped and started automatically as part of an Exchange 2007 or 2010 server agent installation.

Chapter 3: Sendmail and Postfix Integration

This section describes how the Milter MTA agent enables CA DataMinder to monitor and control emails transiting through Sendmail and Postfix servers.

Note: You must enable CA DataMinder email server agents in order to start email processing. By default, the agents are disabled when first deployed.

This section contains the following topics:

About the Milter MTA Agent

CA DataMinder can integrate with Sendmail and Postfix. Sendmail is a mail transfer agent (MTA) originating from the open source and Unix communities. Postfix is intended to be an easy-to-administer, secure alternative to the Sendmail MTA. Both are used to route and deliver email and provide additional services such as protecting organizations from unwanted messages (spam).

CA DataMinder integration with Sendmail and Postfix is implemented through its Milter MTA agent; this agent can reside directly on the Sendmail and Postfix server or on a remote Linux machine, connecting to the email server via a socket.

The Milter MTA agent uses the Sendmail Mail Filter API (milter) to access emails as they transit through Sendmail or Postfix. The milter allows third-party applications access to emails as they are processed in order to filter meta-information and content.

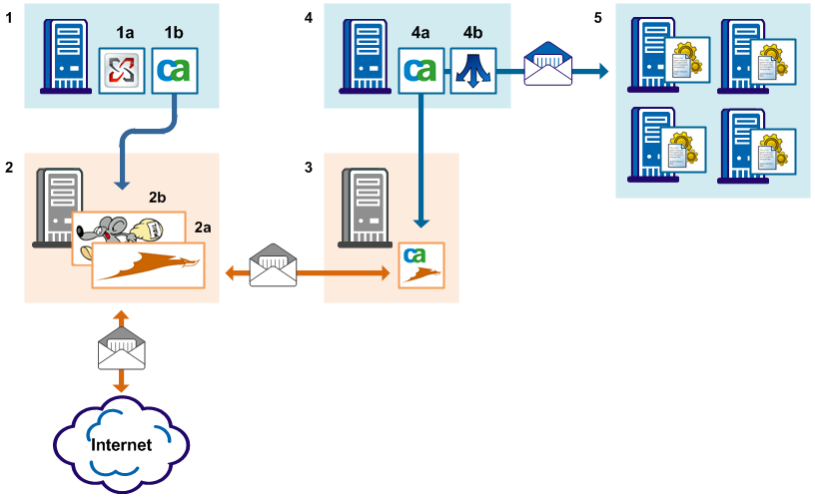
Deployment Architecture

The diagram below summarizes the deployment architecture for CA DataMinder integration with Sendmail or Postfix. This integration is enabled through the CA DataMinder Milter MTA agent. This agent can reside directly on the Sendmail or Postfix email server or, as in this example, on a separate Linux machine.

To enable communication between Unix or Linux machines and the CA DataMinder policy engines running on Windows servers, the Milter MTA agent uses the CA DataMinder Socket API to call the CA DataMinder External Agent. The External Agent in turn establishes a connection with a local policy engine hub. The hub then distributes the Sendmail or Postfix emails to policy engines for processing. Alternatively, the External Agent can pass e-mails directly to a local policy engine (not shown in the diagram below).

Note: Milters are email filtering programs supported by Sendmail and Postfix. They are widely used to filter incoming emails, for example, to block spam.

Note: The External Agent API enables third party applications to pass messages to CA DataMinder for policy processing.



Sendmail and Postfix integration

In a typical Sendmail or Postfix deployment:

1 Emails sent internally transit through an Exchange or Domino server (**1a**), and may be processed by a CA DataMinder server agent (**1b**).

2 The Sendmail or Postfix MTA operates at the Internet boundary. Emails leaving the company or arriving from an external source are processed by Sendmail or Postfix. These messages are forwarded to the CA DataMinder Milter MTA agent (**3**).

3 Hosted on a Linux machine, the Milter MTA agent uses the Socket API (**4a**) to pass emails to CA DataMinder.

4 Hosted on a Windows machine, the Socket API (**4a**) sends Sendmail or Postfix emails to a local policy engine or hub (**4b**). This example shows a local hub. (When installed with the Socket API, policy engine hubs are technically known as Remote PE Connectors).

5 The hub then distributes emails to policy engines for processing. The results of any policy processing are returned via the hub to the Milter MTA agent, which in turn relays any resulting actions back to the Sendmail or Postfix server.

Requirements for the Milter MTA Agent

The Milter MTA agent has been tested using the Linux and Sendmail combinations listed below. The Milter MTA agent may work with other combinations, but these have not been tested.

Linux

The Milter MTA agent has been tested on the following systems:

- Ubuntu 6.04 and 8.04
- Red Hat Enterprise Linux 4 Server.

Sendmail

The Milter MTA agent supports Sendmail 8.13.0 or later (but see below).

We have also successfully connected a Milter MTA agent to Sendmail running on a remote Solaris server.

Milter API Library requirements

During installation, the Milter MTA agent checks for the presence of the Milter API shared library, libmilter.so. If this library is not present, the Milter MTA agent uses a statically compiled libmilter (version 8.13.8), compatible with all versions of Sendmail from 8.13.0 onwards.

But if libmilter.so is already present, the Milter MTA agent uses this library. However, due to a known issue with the Milter API, the agent may not work with libmilter.so 8.14.0 or 8.14.1 (this issue is fixed in version 8.14.2 and later).

Postfix

The Milter MTA agent integrates with any version of Postfix that supports the Sendmail Milter API as specified under http://www.postfix.org/MILTER_README.html.

Applying Policy Triggers to Sendmail and Postfix Emails

When policy engines process emails captured by the Milter MTA agent, they apply Outgoing email policy triggers. But be aware of the following limitations:

Server-side warnings are not supported

For emails detected by the Exchange server agent that generate a warning event, CA DataMinder can automatically send a warning email to the sender. However, such server-side warnings are not supported for e-mails detected by the Milter MTA agent.

Note: Warning emails **are** sent if the Milter MTA agent blocks an email.

Moving email recipients to Bcc list

For emails detected by the Milter MTA agent, you cannot differentiate between internal and external recipients when moving addresses the Bcc field. That is, you can move all or none of the recipients. (Normally, Address Modification settings in outgoing email control actions let you move all recipients, or only external recipients, to the Bcc field.)

How to Deploy the Milter MTA Agent

This section describes how to deploy the Milter MTA agent and the policy engine hub.

The key deployment tasks are:

1. Verify the Milter MTA agent requirements.
2. Deploy your policy engines.

   We recommend you do this before deploying your email server agent. This is described in Deploy policy engines.
3. Deploy the Socket API and a policy engine hub.

   The Milter MTA agent uses these components to pass Sendmail or Postfix emails to policy engines for processing.
4. Deploy the Milter MTA agent.
   a. Specify the Milter user account.
   b. Configure Sendmail or Postfix.
   c. Install the Milter MTA agent.
   d. Configure the Milter MTA agent.
5. Turn on Sendmail or Postfix integration.

   To enable the agent and turn on integration, edit wgnmilter.conf (see page 74).

   Important! You must enable CA DataMinder email server agents in order to start email processing. By default, the agents are disabled when first deployed.

More information:

Policy Engine Hubs (see page 91)
Configure Postfix (see page 66)
Create User for Milter MTA Agent (see page 64)

Deploy Policy Engines

For installation and configuration instructions, see the Policy Engines chapter in the *Platform Deployment Guide*.

In particular read the instructions for specifying the PE domain user. You must specify this user account when you install a policy engine hub.

Deploy the Socket API and a Remote PE Connector

The Milter MTA agent uses the CA DataMinder Socket API to pass Sendmail or Postfix emails to:

- A local policy engine, or
- A Remote PE Connector. This a policy engine hub running on the same machine as the External Agent; this hub then distributes emails to remote policy engines for processing.

You install the Socket API and a Remote PE Connector when you install the External Agent API. In particular, when you install the Socket API you must specify which port number to use for the socket connection between the Milter MTA agent and the External Agent. You will reference this port number when you install the Milter MTA agent.

You install the Socket API and hub with the CA DataMinder Integration Agents installation wizard.

To install the Socket API and a policy engine hub

1. Find setup.exe in the root of your CA DataMinder distribution image. Run setup.exe to launch the CA DataMinder installation wizard.

   The Installation Type screen opens.

2. Click Advanced Installation.
3. In the Advanced Install Options screen, choose Server Agents and then click Install.

   This launches the Integration Agents installation wizard in a separate window.

4. In the Integration Agents installation wizard, navigate to the Customer Setup screen.
5. In the Custom Setup screen, choose External Agent API.
6. In the External Agent API Configuration screen, choose whether to install a Remote Policy Engine Connector (that is, a policy engine hub) and the Socket API. If you install the:

   - **Remote Policy Engine Connector**, you must programmatically configure the External Agent API output destination to be a local hub. Full details are in the *External Agent COM API Specification*.
   - **Socket API**, you can use socket connections to call the External Agent API from a remote location, including from a non-Windows system. For example the CA DataMinder Network uses the Socket API to analyze traffic leaving or entering the corporate network from the internet. By default, the Socket API automatically listens on port numbers **8538** and **8359**.

7. If you chose to install a Remote Policy Engine Connector in step 6, specify the PE domain user in the Policy Engine Hub Configuration screen.
8. In the final wizard screen, click Install to start the file transfer.
9. When the file transfer is complete, the wizard installs the necessary DLLs and registry values to the External Agent API machine.
    a. It creates a new CA DataMinder installation folder (unless this folder exists already, containing Event Import). It then installs Wgnrdi.dll to the \client subfolder of this CA DataMinder folder.
    b. It creates the following registry key. You may need to edit registry values in this key.

        ```
        HKEY_LOCAL_MACHINE\Software\ComputerAssociates\CA DataMinder
          \CurrentVersion \External Agent API
        ```

For full details about the External Agent API, see the *Archive Integration Guide.*

Configure the PE Hub

This is described in Configure the policy engine hub (see page 98).

Create User for Milter MTA Agent

The Milter MTA agent must run as a designated user (it will not run as root). You must specify this user when you install the Milter MTA agent. This is your 'Milter user'.

You can either specify an existing user as your Milter user or you can create a new user. If you create a new user, you must do so before you install the Milter MTA agent.

To create a new Milter user

Run this command as root:
`useradd <user name>`

Where <user name> is the name of your Milter user. For example:
`useradd miltuser`

More information:

Install the Milter MTA agent (see page 65)

Install the Milter MTA agent

When you install the Milter MTA agent, you are prompted for details about the Milter user, installation directory and socket connection.

To install the Milter MTA agent

1. As root, run this shell command to extract the installation files:

   `<path>/install.sh`

   Where <path> specifies the mount point of the CA DataMinder distribution media containing the install.sh file. For example:
   `<mount>/lin_i/WgnMilter/install.sh`

   Where <mount> is the mount point.

2. When the file extraction is complete, install.sh prompts you for the following details.

3. **Sendmail or Postfix:** Specify whether you are installing an integration agent for Sendmail or Postfix (options 1 and 2 respectively).

4. **Existing mail agent user:** Specify the 'Milter user' you chose or created, (for example, miltuser). The Milter MTA agent runs as this user.

5. **Installation directory:** Specify the directory where you want to install the Milter MTA agent. For example:
   `/opt/milt`

6. **Socket identifier for Sendmail:** Specify the socket that will be used for communication between Sendmail and the Milter MTA agent. Note that this socket is created and deleted when you start and stop the Milter MTA agent.

   Important! This identifier **must** match the socket specified in sendmail.mc.

   Specify a local socket if you are installing the Milter MTA agent directly onto the Sendmail or Postfix server. Specify an Internet socket if you are installing the Milter MTA agent on a separate Linux machine.

   - **Local socket:** The socket syntax takes this format:

     `local:/opt/milt/wgnmilter.sock`

     It defaults to the installation directory you specified in step 5 (in this case, /opt/milt), but you can specify an alternative location if required.

   - **Internet socket:** The syntax takes this format:

     `inet:CA Portal@<agent host server>`

     For Internet sockets, the specified CA Portal number must not be used by another application. You can identify the Milter MTA agent host server by IP address or name (such as milter.my.com). For example:

     `inet:8600@milter.my.com`

7. **IP address of Socket agent host machine:** Enter the IP address of the machine hosting the CA DataMinder Socket API.
8. **Socket agent port number:** Enter the port number for the listening port on the Socket API host server. Or you can accept the default port (8538).

   Note: Press Enter to accept the default port number.
9. **Configure a secondary Socket agent:** If required, you can specify a secondary Socket API on a separate machine to ensure high availability. If your primary Socket API (step 7) becomes unavailable for any reason, the Milter MTA agent will automatically switch to the secondary Socket API.

   Type y or n to indicate whether or not you want to configure a secondary Socket API. If you type y, the installation script prompts you for the IP address and port number on the machine hosting the secondary Socket API.
10. install.sh now lists a summary of the Milter MTA agent details. If you need to amend any details, you can do so now. Otherwise, confirm the details to begin the file transfer.

    In particular, the install.sh script installs the following items to the installation directory you specified in step 5:

    - wgnmilter.conf. This is the Milter MTA agent configuration file. It is installed into the installation directory you specified in step 5. You will need to edit this file to configure the agent.
    - Scripts to automatically start and stop the Milter MTA agent when the host machine starts up or shuts down.
    - RunWgnMilter script. Run this script to update the Milter MTA agent with changed configuration details.
    - Uninstall script. Run this script to uninstall the Milter MTA agent.

Configure Postfix

For Postfix configuration instructions, please contact CA Technical Support.

Configure Sendmail Integration

Before you install the Milter MTA agent, you must add a line to the sendmail.mc file on the Sendmail server. This line identifies the socket used for communication between Sendmail and the Milter MTA.

Note: The socket identifier in sendmail.mc file **must** match the socket parameter specified in wgnmilter.conf.

More information:

Example Socket Commands (see page 69)
Wgnmilter.conf Parameters (see page 70)

Milter MTA - Command Syntax

The command syntax is:

```
INPUT_MAIL_FILTER ('WgnMilter','S=<socket>[,<comm_fail>][,<timeouts>]')dnl
```

where

<socket>

Locates the socket used to communicate with the Milter MTA agent. It takes these formats:

Either: local:<path>/wgnilter.sock

Or: inet:CA Portal@<Agent host server>

Use the local: format if the Milter MTA agent is installed directly on a Sendfix or Postmail server. Otherwise, use inet: to specify an Internet socket; note that the specified CA Portal number must not be used by another application.

<comm_fail>

Is an optional parameter that determines how Sendmail handles emails when the Milter MTA agent is not running or when there is a communication failure with the Milter MTA agent. It can be set to F=R, F=T or omitted entirely.

If the entire parameter is omitted, emails transit through Sendmail without intervention.

If set to F=R then Sendmail rejects all emails arriving through the SMTP connection from Exchange or Domino. Sender of rejected emails eventually receives an ‘undeliverable message’ notification.

If set to F=T then Sendmail temporarily fails any e-mail arriving through the SMTP connection. The source email server (Exchange or Domino) automatically resends the emails at predefined intervals; if the Milter MTA agent has restarted when the e-mail is resent, it is processed by CA DataMinder as normal.

<timeouts>

Is an optional parameter for changing the default time-outs associated with the Milter MTA agent. It can take any combination of the following parameters, in any order:

`T=C:<t>;E:<t>;S:<t>;R:<t>`

C

Is the connection timeout from Sendmail to the Milter MTA agent. This defaults to 5 minutes.

E

Is the overall timeout for an email. This covers the period from when an email is first submitted to the Milter MTA agent until the results of policy processing are returned to Sendmail or Postfix. This timeout defaults to 5 minutes.

To ensure that Sendmail does not resubmit e-mails unnecessarily to the Milter MTA agent, this E timeout **must** be longer than the max-time-per-mail timeout (see max-time-per-mail=<number>) specified in wgnmilter.conf.

S

Is the timeout for sending individual data packets to the Milter MTA agent. This defaults to 10 seconds.

R

Is the timeout for receiving a reply from the Milter MTA agent. This defaults to 10 seconds.

<t>

Is the timeout value, suffixed with m or s to indicate minutes or seconds. For example, 5m is a five minute timeout; 90s is a 90 second timeout.

Note: You may need to lengthen the default time-outs to allow time for policy processing.

More information:

Example Socket Commands (see page 69)
Wgnmilter.conf Parameters (see page 70)

Example Socket Commands

Local Socket

The following example command specifies a local socket in the /opt/milt directory. It also overrides the default time-outs; if there is a communication failure between Sendmail or Postfix and the Milter MTA agent, the Send timeout is set to five minutes while the Receive and Overall time-outs are set to 10 minutes. If the Milter MTA agent is not running, Sendmail rejects e-mails arriving from an Exchange or Domino server.

```
INPUT_MAIL_FILTER
('WgnMilter',
 'S=local:/opt/milt/wgnmilter.sock,
  F=R,
  T=S:5m;R:10m;E:10m'
)dnl
```

Internet Socket

You must specify an Internet socket if the Milter MTA agent runs on a different machine from the Sendmail server. For this example, the Milter MTA agent is hosted on milter.my.com. and uses port 8600 to communicate with the Sendmail server. The time-outs are the same as for the 'local socket' example. If the Milter MTA agent is not running, e-mails arriving from an Exchange or Domino server are pass through Sendmail without intervention.

```
INPUT_MAIL_FILTER
('WgnMilter',
 'S=inet:8600@milter.my.com,
  T=S:5m;R:10m;E:10m'
)dnl
```

Configure the Milter MTA Agent

You supply basic configuration details when you install the Milter MTA agent. But after installation, you can fine-tune the agent configuration by manually editing parameters in wgnmilter.conf; this configuration file is created in the installation directory for the Milter MTA agent.

More information:

Wgnmilter.conf Parameters (see page 70)
Install the Milter MTA agent (see page 65)

Wgnmilter.conf Parameters

This section lists the configuration parameters supported in wgnmilter.conf.

Note: These parameters are **not** case-sensitive.

enable-integration=0 or 1

Defaults to 0 (zero). This parameter turns integration on or off. Set this parameter to:

0 to turn off integration. This disables the Milter MTA agent, so that emails are allowed to transit through Sendmail or Postfix without intervention from CA DataMinder.

1 to turn on CA DataMinder integration with Sendmail or Postfix.

milter-socket=<socket>

This mandatory parameter specifies the socket that the Milter MTA agent uses to communicate with the Sendmail or Postfix server.

Important! This parameter **must** match the socket specified in sendmail.mc.

This parameter can specify a local socket or an Internet socket. Specify a local socket if you are installing the Milter MTA agent directly onto the Sendmail or Postfix server. Specify an Internet socket if you are installing the Milter MTA agent on a separate Linux machine.

- **Local socket:** The socket syntax takes this format:

  ```
  local:/opt/milt/wgnmilter.sock
  ```

- **Internet socket:** The syntax takes this format:

  ```
  inet:CA Portal@<Agent host server>
  ```

  For Internet sockets, the specified CA Portal number must not be used by another application. You can identify the Milter MTA agent host server by IP address or name (such as milter.my.com). For example:

  ```
  inet:8600@milter.my.com
  ```

primary-policy-ipaddress=<IP address>

This mandatory parameter specifies the IP address of the machine hosting the Socket API. It corresponds to installation step 7 of Install the Milter MTA agent.

primary-policy-port=<port number>

This mandatory parameter specifies the port number for the listening port on the Socket API host machine. It corresponds to installation step 8 of Install the Milter MTA agent.

secondary-policy-ipaddress=<IP address>

This optional parameter specifies the IP address of the machine hosting the secondary Socket API. It corresponds to installation step 9 of Install the Milter MTA agent.

secondary-policy-port=<port number>

This optional parameter specifies the port number for the listening port on the machine hosting the secondary Socket API. It corresponds to installation step 9 of Install the Milter MTA agent.

email-failure-mode=<action>

This parameter can be set to Delete, Allow, or Mark. It defaults to Allow. It specifies how the Milter MTA agent handles event failures. If set to:

- Delete, the Milter MTA agent deletes the e-mail without notifying the sender.
- Allow, the Milter MTA agent allows the email to transit through Sendmail or Postfix without intervention.
- Mark, the Milter MTA agent allows the email to transit through Sendmail or Postfix without intervention, but marks it as if it had been processed normally. This prevents a downstream CA DataMinder agent from re-processing the email.

Email failures can occur when:

- The hub's HighWaterMarkEventCount or HighWaterMarkMB thresholds are exceeded and the HubFailureMode flags all subsequent emails as failures
- The GlobalEventTimeoutSeconds hub timeout expires
- There is a general failure when the policy engine analyzes the email.

sys-log-level=0 through 7

Defaults to 5. This parameter determines the logging level for email processing. Log messages are written to syslog. For example, you can configure the Milter MTA agent to only log alerts and emergency messages—but see the note below. The supported logging levels are:

0 Emergency

1 Alert

2 Critical

3 Errors

4 Warnings

5 Notices

6 Information

7 Debug

Note: Logging levels are cumulative so, for example, logging level 2 causes critical messages, alerts and emergency messages to be written to syslog.

diagnostic-folder=<path>

This locates the folder where diagnostic files are written to. For example, set this parameter to:

```
diagnostic-folder=/opt/WgnMilter/diag
```

Be aware that if this parameter (diagnostic-folder) is not set, no diagnostic files are created, even if create-eml is set (see below).

create-eml=0, 1 or 2

Defaults to 0. This parameter is provided for diagnostic purposes. It specifies whether to create diagnostic files containing the emails and associated data for emails processed by the Milter MTA agent.

These diagnostic files comprise an EML file, containing the 'raw' MIME content of the email, and an SMTP file, containing the sender and recipient details. Any diagnostic files created are saved in the diagnostic-folder (see above).

If create-eml is set to:

0, diagnostic files are never created.

1, diagnostic files are always created for each e-mail processed by the Milter MTA agent.

2, diagnostic files are only created on error. For example, this can happen if an event times out while waiting to be processed by a policy engine.

max-time-per-mail=<number>

Defaults to 600; the minimum permitted value is 15. This parameter specifies the maximum processing time (in seconds) for each email.

This timeout covers the period from when an email is first received by the Milter MTA agent until the results of policy processing are returned to Sendmail or Postfix.

To ensure that Sendmail does not resubmit e-mails unnecessarily to the Milter MTA agent, this parameter **must** specify a timeout shorter than the 'E' timeout specified in sendmail.mc.

sender-address-inclusion-filters=<address list>

Defaults to *. This value specifies a comma separated list of SMTP addresses for the Milter MTA agent to filter against. For example, if you only want to monitor emails sent from the unipraxis.com domain, you need only set this value to 'unipraxis'. By default, the Milter MTA agent monitors all emails passing through Sendmail or Postfix.

If the sender's email address does not match any item in this list, the Milter MTA agent disregards the email and allows the email to transit through Sendmail or Postfix without intervention.

Similarly, if you want to test that integration with Sendmail or Postfix is working correctly before you go live, you can specify the full address of a test user. This ensures that if there is problem with the integration, no other users will be affected.

Note that * and ? wildcards are supported. A * wildcard will match any sequence of zero or more digits, letters or punctuation characters. For example, a *unipraxis* filter will match spencer@sales.unipraxis.com. A ? wildcard will match any digit, letter or punctuation character, for example, spen?er matches Spencer or Spenser.

enterprise-dns-list=<domain list>

Defaults to empty, but see 'Updating an empty list' below. This parameter specifies a list of DNS domains that you want to be considered as a single enterprise. It is typically used in conjunction with smtp-dns-hostname (see below). Its purpose is to simplify the method for ensuring that emails are not reprocessed needlessly by consecutive CA DataMinder server agents.

Note: Values in this list can only contain ASCII characters, without spaces or control characters (for example, tab spaces).

How does this work? The local Milter MTA agent assumes each of the domains listed in enterprise-dns-list has its own CA DataMinder agent (that is, an Exchange or Domino server agent, or a Milter MTA agent), and that any email arriving from a listed domain has already been processed by CA DataMinder and does not need reprocessing.

In technical terms, when a *remote* Milter MTA agent processes an email, it writes the DNS domain of the Sendmail or Postfix server to the email's MIME tag. Or, if the smtp-dns-hostname parameter has been configured, this DNS domain is written to the MIME tag instead.

When the *local* Milter MTA agent receives this e-mail, it checks the MIME tag. If the source domain matches a domain in enterprise-dns-list, the server agent does not reprocess the email.

Updating an empty list

While this list is empty, the DNS domain of the local Sendmail or Postfix server is implied. That is, the local Milter MTA agent does not reprocess e-mails arriving from this domain.

Note: For examples of how the equivalent registry value is used for the Exchange or Domino server agent.

smtp-dns-hostname=<dns host>

Defaults to empty. This parameter is always used in conjunction with enterprise-dns-list (see above). Its purpose is to simplify the method for ensuring that emails are not reprocessed needlessly by consecutive Milter MTA agents.

Note: This parameter must be set to a valid DNS name that complies with RFC naming conventions.

smtp-dns-hostname specifies a single DNS domain that is written to the email's MIME tag after it has been processed by the local Milter MTA agent. If set, this parameter overrides the DNS domain of the Sendmail or Postfix server in the MIME tag.

To use this parameter as intended, you need to set smtp-dns-hostname to the same DNS domain (for example, UNIPRAXIS.COM) for all your Milter MTA agents. You can optionally include this domain in the enterprise-dns-list domain list. Now, when any Milter MTA agent receives an e-mail tagged as coming from UNIPRAXIS.COM, it knows that policy has already been applied to the email and so does not reprocess it.

Note: For examples of how the equivalent registry value is used for the Exchange or Domino server agent.

More information:

Prevent Repeat Processing by Server Agents in Multiple Domains (see page 117)
Configure Sendmail Integration (see page 66)
Install the Milter MTA agent (see page 65)

Turn On Sendmail and Postfix Integration

Important! You must enable CA DataMinder email server agents in order to start email processing. By default, the agents are disabled when first deployed.

The Milter MTA agent uses a single configuration parameter in wgnmilter.conf to turn on email integration with Sendmail or Postfix. After editing wgnmilter.conf, you must update the Milter MTA agent with the new configuration.

To turn on email integration

1. Edit wgnmilter.conf.

   This configuration file is in the installation folder for the Milter MTA agent.

2. Change the enable-integration line to:
   `enable-integration=1`

3. Run *one* of the following commands:
   `RunWgnMilter –u`
   `RunWgnMilter ––updateconfig`

4. Start the Milter MTA agent.

   a. Find the wgnmilter script installed with the Milter MTA agent in the /init.d folder.

   b. Run the wgnmilter script, specifying 'start'.

      For example, on Red Hat machines the start command takes this format:
      `/etc/init.d/wgnmilter start`

   Email processing is enabled.

   Note: You can optionally configure Sendmail (or Postfix) to only send emails when the Milter MTA agent is running.

To turn off email integration

1. Edit wgnmilter.conf.

2. Change the enable-integration parameter to:
   `enable-integration=0`

3. Run *one* of the following commands:
   `RunWgnMilter –u`
   `RunWgnMilter ––updateconfig`

   Email processing is disabled. The Milter MTA agent allows all emails to transit through Sendmila or Postfix without intervention.

4. (Optional) Stop the Milter MTA agent. If you stop the Milter MTA agent completely, the agent stops processing emails.

   a. Find the wgnmilter script (see above).

   b. Run the wgnmilter script, specifying 'stop'.

      For example, on Red Hat machines the stop command takes this format:
      `/etc/init.d/wgnmilter stop`

To configure Sendmail to only send emails when the Milter MTA agent is running

1. Open the Sendmail configuration file sendmail.mc.

   Find this file in the Sendmail configuration directory. This directory is normally /etc/mail.

2. Add *one* of the following options to the line that specifies the socket connection.

   F=R

   Specifies that Sendmail rejects emails arriving from an Exchange or Domino server.

   F=T

   Specifies that Sendmail temporarily fails emails arriving from an Exchange or Domino server.

   Position the F option inbetween the S and T options. For example:

   ```
   INPUT_MAIL_FILTER(`WgnMilter',
   `S=inet:9999@ux-mailsvr-w2k3.unipraxis.com,
    F=R, T=S:4m;R:10m;E:10m')dnl
   ```

Set Up SecureMail Integration

You can configure CA DataMinder to detect, analyze, and apply policy to emails encrypted by Voltage SecureMail. Specifically, the following agents can detect and apply policy to emails encrypted by Voltwage SecureMail:

- Exchange server agent
- IIS SMTP server agent
- Milter MTA agent (for Sendmail and Postfix email servers)

To set up SecureMail integration, you must edit the registry on your policy engines and set a shared secret. Policy engines use the shared secret to establish secure connections to the Voltage SecureMail server. Details are in the Voltage SecureMail Integration chapter.

More information:

Voltage SecureMail Integration (see page 79)

Uninstall the Milter MTA Agent

Do the following if you want to uninstall the Milter MTA agent. Uninstalling will not delete wgnmilter.conf or the diagnostic folders and files.

To uninstall

1. Verify that you are the root user.
2. Find the uninstall script in the installation directory.
3. Run the uninstall command as root.

   ```
   uninstall
   ```

More information:

Install the Milter MTA agent (see page 65)

Chapter 4: Voltage SecureMail Integration

This section describes how to integrate CA DataMinder email server agents with Voltage SecureMail.

This section contains the following topics:

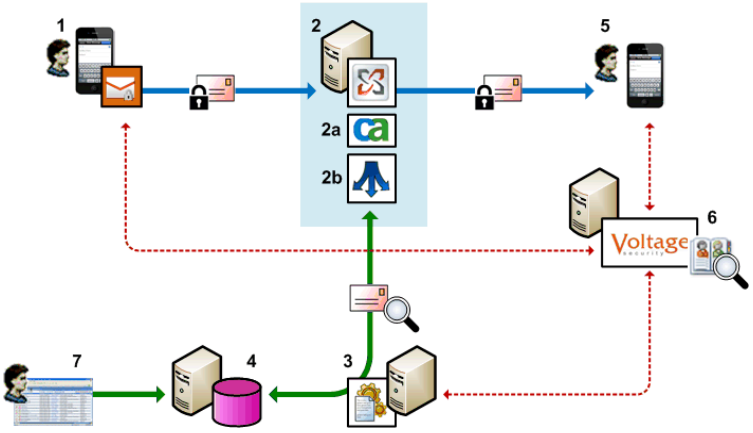
Overview

You can configure CA DataMinder to detect, analyze, and apply policy to emails encrypted by Voltage SecureMail.

How does the integration work? CA DataMinder intercepts Voltage-encrypted emails passing through an email server and passes copies of these emails to a CA DataMinder policy engine. The policy engine establishes a secure connection to the Voltage SecureMail server, which provides the policy engine with an unencrypted version of the email. The policy engine can then apply policy triggers to the email as normal. When policy processing is complete, the policy engine calls back to the email server agent. The callback instructs the email server agent to either block the encrypted email or allow it to continue.

Note: The original encrypted email remains on the email server until policy processing is complete.

The key components are shown below. For simplicity, this diagram shows the Exchange server agent passing encrypted emails to a single policy engine.



Example deployment architecture: Exchange server agent integration with Voltage SecureMail

A employee (**1**) sends a secure email from their mobile device. The device is running the SecureMail app. The app connects to the Voltage SecureMail server (**6**) to authenticate the sender and encrypt the email.

The encrypted email passes through the Exchange server (**2**). It is intercepted by the Exchange server agent (**2a**) and forwarded to the PE hub (**2b**). The PE hub distributes a copy of the email to a policy engine (**3**).

The policy engine establishes a secure connection to the Voltage SecureMail server (**6**), which sends back an unencrypted version of the email.

The policy engine applies Outgoing Email triggers to the email and calls back to the Exchange server agent with the results of the policy processing (for example, 'block the email'). The resulting email event is replicated to the CMS (**4**).

If policy processing allows the email to continue, the original encrypted email is forwarded to the recipient (**5**). To decrypt the email, the recipient authenticates themself to the Voltage SecureMail server (**6**).

A reviewer (**7**) can search for email events in the iConsole. Unencrypted versions of emails encrypted by SecureMail are available to reviewers and are flagged as 'Secure' in the iConsole.

Requirements

Note the following requirements for CA DataMinder integration with Voltage SecureMail:

Voltage SecureMail

When you enable integration with Voltage SecureMail, CA DataMinder policy engines can access unencrypted versions of emails that were encrypted by Voltage SecureMail clients or the Voltage Zero Download Messenger.

CA DataMinder integrates with the following versions of Voltage SecureMail:

- Voltage SecureMail Mobile Edition
- Voltage SecureMail Cloud Standard Edition
- Voltage SecureMail Cloud Enterprise Edition

Voltage SecureMail Server

You must supply your policy engines with connection details and a shared secret for the Voltage SecureMail web service

Voltage SecureMail Gateway

(Optional) CA DataMinder integration with Voltage SecureMail does not directly require a SecureMail Gateway. However, if you use the CA DataMinder Quarantine Manager to quarantine suspect emails, you must deploy a SecureMail Gateway on your network. The SecureMail Gateway ensures that emails released from quarantine are re-encrypted before they are forwarded to external recipients.

Which agents can detect SecureMail-encrypted emails?

The following CA DataMinder agents can detect and apply policy to emails encrypted by Voltage SecureMail:

- Exchange server agent
- IIS SMTP server agent
- Milter MTA agent (for Sendmail and Postfix email servers)
- CA DataMinder Network (formerly known as the NBA). This agent can detect and apply policy to SMTP emails encrypted by SecureMail.

Policy engines and SSL certificates

CA DataMinder policy engines use the Secure Sockets Layer protocol (SSL) to establish a secure connection to the Voltage SecureMail web service. To ensure that the policy engine and SecureMail web service trust each other, each policy engine must hold a copy of the root certificate that was used to generate the SecureMail certificate.

More information:

Establish an SSL Connection to the SecureMail Web Service (see page 87)

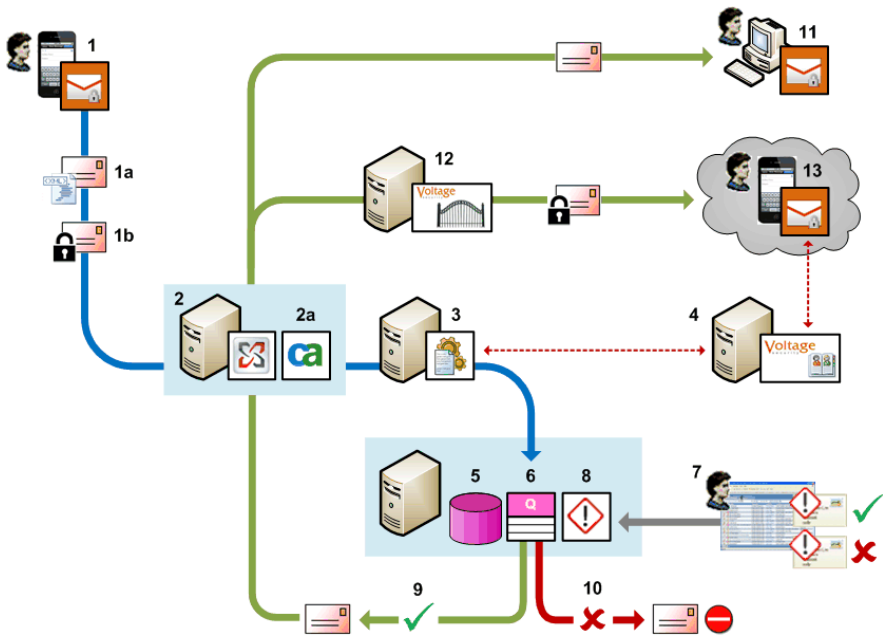
SecureMail Integration and Quarantine Manager

CA DataMinder is able to quarantine emails until they have been approved by an appropriate representative.

If you want to quarantine emails encrypted by Voltage SecureMail, you must deploy the CA DataMinder Quarantine Manager and a Voltage SecureMail gateway.

In the example below, a user sends a secure email to an internal recipient and an external recipient. CA DataMinder detects the email, applies policy, and quarantines the email. When the email is released from quarantine, the version addressed to an external recipient is re-encrypted by the Voltage SecureMail gateway before that version is delivered. The version addressed to an internal recipient is delivered directly and is not re-encrypted.

Note: For simplicity, this diagram omits a policy engine hub and shows the Exchange server agent passing emails to a single CA DataMinder policy engine.



Example: SecureMail integration and quarantined emails on an Exchange server

An employee (**1**) sends a secure email from their mobile device. If they use a Voltage SecureMail clientless solution, the email is not encrypted yet, but contains an x-header to indicate that encryption is required (**1a**). If they use a Voltage SecureMail end-to-end solution, the email is encrypted before it leaves the mobile device.

When the secure email passes through the Exchange server (**2**), the Exchange server agent (**2a**) intercepts the email and passes it to a policy engine (**3**) for analysis.

The policy engine establishes a secure connection to the Voltage SecureMail server (**4**), which sends back an unencrypted version of the email. The policy engine analyzes the email and applies a Quarantine action. A decrypted version of the email is saved in the CMS database (**5**). The email is also saved in a quarantine queue (**6**).

iConsole reviewers can search for quarantined emails and release or reject them (**7**).

The Quarantine Manager (**8**) regularly checks the quarantine queue and forwards released emails to their intended recipients (**9**). At this stage, emails released from quarantine are unencrypted but include an 'encryption request' x-header. Rejected emails are not forwarded (**10**).

When released emails pass through the Exchange server, any emails addressed to internal recipients are sent directly, unencrypted, to the recipient (**11**).

Emails addressed to external recipients are routed through a Voltage SecureMail Gateway (**12**). The Gateway detects the 'encryption request' x-header and re-encrypts the email before sending it to the intended recipient (**13**). To decrypt the email, the recipient must authenticate themself to the Voltage SecureMail server (**4**).

Set Up SecureMail Integration

See the following sections for full details. Briefly, you must:

1. Edit registry values on the policy engine to identify the Voltage SecureMail server and enable integration with SecureMail.
2. Set a shared secret.

   The policy engine uses the shared secret to establish secure connections to the Voltage SecureMail server.
3. Configure the Voltage SecureMail server.

   You must add a new component authentication method and enable the Web Services API.
4. Establish an SSL connection between the policy engine and the SecureMail web service.
5. (Optional) Define a rule on your Voltage SecureMail Gateway to detect emails containing an encryption request x-header.

More information:

Configure the Policy Engine Registry Values (see page 85)
Set a Shared Secret for the Voltage SecureMail Server (see page 86)
Establish an SSL Connection to the SecureMail Web Service (see page 87)
Define Rule for Voltage SecureMail Gateway (see page 89)

Configure the Policy Engine Registry Values

To complete the setup for integration with Voltage SecureMail, you must edit registry values on the policy engine host machine. These values enable the policy engine to connect to the Voltage SecureMail server. To configure access to the Voltage SecureMail server, locate the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
    \CurrentVersion\Security Providers\Voltage
```

Within the specified registry key, edit the following registry values:

Enabled

Type: REG_DWORD

Data: Defaults to 0. To enable integration with Voltage SecureMail, set this registry value to 1. This ensures that policy engines attempt to decrypt encrypted emails detected by CA DataMinder email servfer agents. Set this registry value to 0 to disable integration.

ServerURL

Type: REG_SZ

Data: Specifies the protocol, server, and optional port number for the Voltage SecureMail web service that the PE uses to decrypt emails. For example:

```
https://voltage-pp-0000.unipraxis.com
https://voltage-pp-0000.unipraxis.com:425
```

You only need to specify the port number if the web service is *not* using a default port. The defaults are port 80 for HTTP and port 443 for HTTPS.

Set a Shared Secret for the Voltage SecureMail Server

When the policy engine receives a copy of the encrypted email, it sends the email plus a shared secret to the Voltage SecureMail server.

The shared secret (in effect, a password) authenticates the policy engine to the Voltage SecureMail server. It also enables the policy engine to establish a secure connection with the Voltage SecureMail server.

In return, the Voltage SecureMail server provides the policy engine with an unencrypted version of the email.

CA DataMinder provides a command line utility, wgncred.exe, to set account credentials for various components. In this case, you must use it to securely store the shared secret that the policy engine uses to access the Voltage SecureMail server.

Wgncred.exe is installed when you install a policy engine. Find wgncred.exe in the \System subfolder of the CA DataMinder installation folder.

You must run wgncred.exe on each policy engine host server.

To set the SecureMail shared secret

1. From a command prompt, run:
   `wgncred -set`

   A list of components is displayed with their corresponding ID numbers and component identifiers.
2. Choose the Voltage SecureMail Web Service component. The component ID is Voltage.
3. Type the password for this component.

To clear the SecureMail shared secret

1. From a command prompt, run:
   `wgncred -clear`

   A list of components is displayed.
2. Choose the Voltage SecureMail Web Service component. The component ID is Voltage.

Establish an SSL Connection to the SecureMail Web Service

CA DataMinder policy engines use the Secure Sockets Layer protocol (SSL) to establish a secure connection to the Voltage SecureMail web service. To ensure that the policy engine and SecureMail web service trust each other, each policy engine must hold a copy of the root certificate that was used to generate the SecureMail certificate.

To install the SecureMail root certificate on your policy engine

Use the Microsoft Management Console to manage root certificates on your policy engine host servers. If required, you can export the root certificate from your SecureMail server and import it onto your policy engine host servers. Exported certificates are saved as files. Copy the file to your policy engine host server and then double-click the file to launch the certificate import wizard.

For further details, search for 'configuring SSL certificates' and 'importing CA and root certificates' in the *Voltage SecureMail Management Console Administrator Guide*.

How does the SSL connection work?

SSL communications between the policy engines and SecureMail web service are encrypted using public/private key encryption.

When you install a Voltage SecureMail server, an SSL certificate is assigned to the SecureMail web service. This certificate was generated from a root certificate. The root certificate is signed by a certificate authority that is trusted by SecureMail (the 'trusted certificate authority').

When the policy engine establishes an SSL connection, it obtains a public key from the same root certificate that was used to generate the SecureMail certificate. A copy of this root certificate must be already installed on the policy engine host server.

Next, the policy engine requests the SecureMail certificate. Because the SecureMail certificate is signed by a certificate authority that the policy engine trusts, the policy engine proceeds with the connection and encrypts the communication using the public key.

The SecureMail web service then uses a private key to decrypt the encrypted communication.

Note: Browsers ship with, and regularly update, a set of certificates signed by trusted certificate authorities to ensure that connections can be verified.

More information:

What is SSL? (see page 88)
About Certificates (see page 88)

What is SSL?

The Secure Sockets Layer protocol (SSL) helps ensure that a network transaction (such as a web request) is only serviced by the intended network host (such as a web site). SSL also prevents transmitted data from being intercepted by a third party. The connection does this by encrypting the traffic using public/private key encryption.

- You obtain a public key via a certificate which is validated against a trusted certificate authority.
- Each client holds a well-known public certificate of an organization that it trusts (the certificate authority). The client then requests the certificate of the server that it needs to connect to. If the server's certificate is correctly signed by a trusted certificate authority, the client proceeds with the connection and negotiates the encrypted communications channel.

Typical SSL applications include online purchasing and webmail, and an increasing number of web sites and applications (such as instant messaging). In particular, the widespread use of social networking sites is a major cause for concern regarding data loss. Your ability to analyze data transmitted from your company network to these external networks is increasingly important.

About Certificates

A certificate is a small file containing data about a website or network host. The certificate is signed to prevent falsification and contains a chain of responsibility (the certification path) that allows a browser or network client to verify the certificate even if the browser or client only has local access to the top-level (or root) certificate in the chain.

Web browsers provide the ability to view the certificate of a website and verify that the certificate is valid. Browsers ship with, and regularly update, a set of Certificate Authority certificates to help ensure that verification can be performed.

Define Rule for Voltage SecureMail Gateway

If you use a Voltage SecureMail clientless solution (also known as 'SecureMail FlagSecure'), emails are not encrypted when they are sent but contain an x-header to indicate that encryption is required. Therefore you must define a rule on your SecureMail Gateway to encrypt the email.

Note: If you use a Voltage SecureMail end-to-end solution, emails are already encrypted before they leave the workstation or mobile device.

You *must* use 'x-voltage: encrypt'

To support CA DataMinder integration with SecureMail, you *must* define a SecureMail Gateway rule to detect the following x-header:

```
x-voltage: encrypt
```

This is the default x-header for SecureMail FlagSecure. *Do not specify a different x-header!*

Note: For details about configuring gateway rules, see the *Voltage SecureMail Management Console Administrator Guide*.

Why is this necessary?

CA DataMinder policy engines can only recognize 'x-voltage: encrypt' x-headers in SecureMail emails. They cannot recognize different x-headers.

CA DataMinder uses this x-header to mark SecureMail emails as 'Encrypted' when a user reviews these emails in the iConsole.

Note: If you use a different x-header, CA DataMinder still applies policy to the email. And the email is still encrypted by the SecureMail Gateway. However, CA DataMinder does not mark the email as 'Encrypted' in the iConsole.

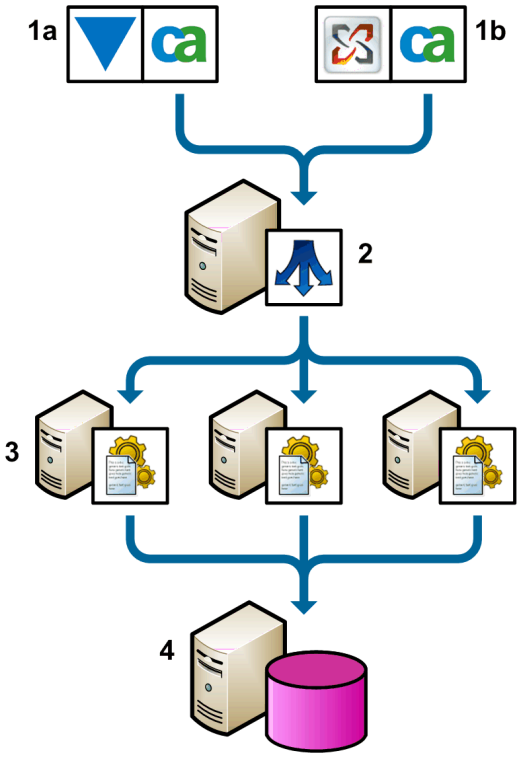
Chapter 5: Policy Engine Hubs

This section contains the following topics:

Policy Engine Hubs Overview

The role of a policy engine hub is to allocate emails to individual policy engines. Policy engine hubs can accept e-mails from email and archive server agents (Exchange, Domino and Enterprise Vault), the Network Boundary Agent (NBA), and also from Event Import.

A policy engine hub handles each email with minimal delay. It distributes email processing across multiple remote policy engines to optimize load-balancing and maximize throughput. It can also handle hardware failures on remote policy engines, seamlessly redistributing events to other policy engines if necessary.

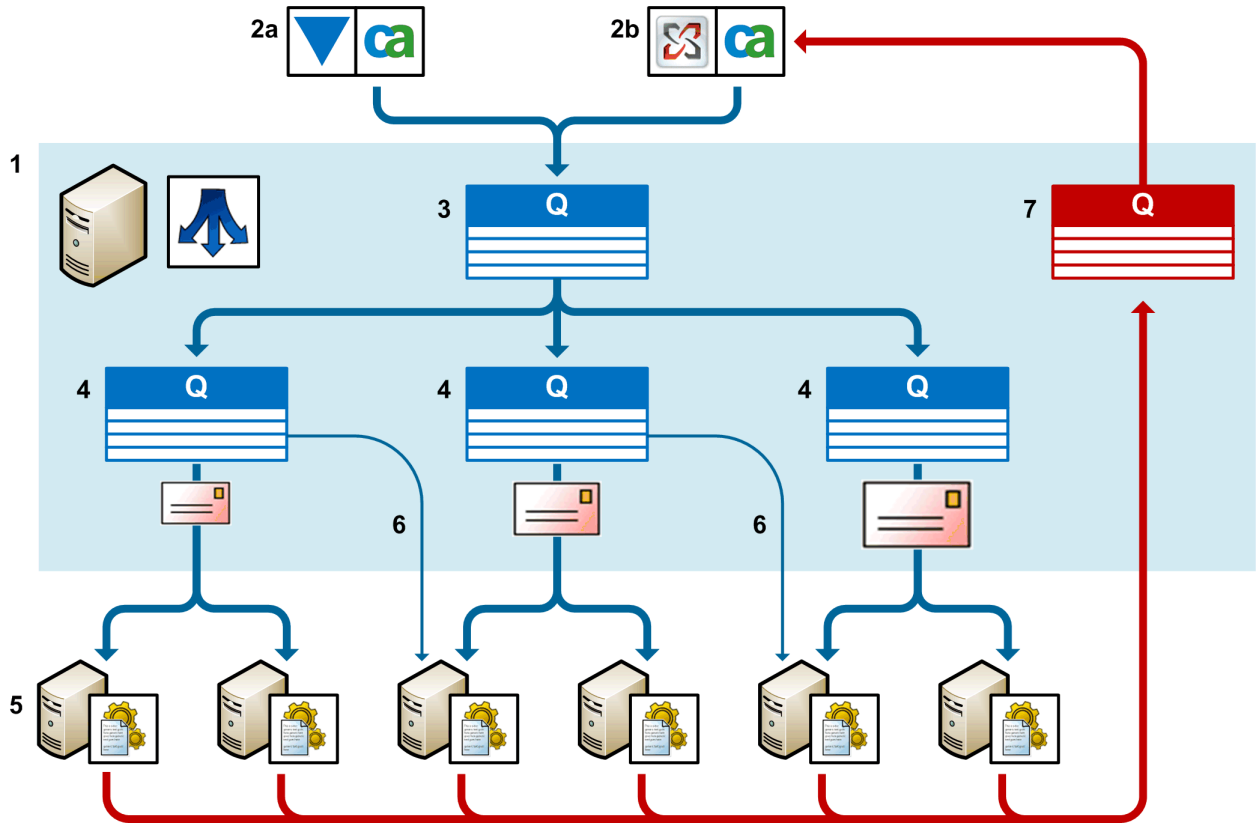


Example policy engine hub deployment

1 Event sources, including CA DataMinder Event Import (**1a**) and e-mail and archive server agents (**1b**) forward emails to the policy engine hub (**2**). The hub distributes e-mails to individual policy engine (**3**) for processing. The resulting events are replicated up to the CMS (**4**).

Policy Engine Hub Architecture

A policy engine hub can have multiple event queues, configurable by email size, with each queue serving multiple policy engines. Distributing emails across multiple queues in this way allows fast-track processing for emails of different sizes. Registry settings on the hub host server define the queue size bands and specify which policy engines are assigned to each queue—see Hub event queues (see page 95) for details. When a policy engine has finished processing an email, it is passed back to the hub before being returned to the source application.



Policy engine hub architecture

A policy engine hub (**1**) can accept emails from various sources, including Event Import (**2a**) and an email server agent (**2b**).

Emails arriving at the hub are added to the input queue (**3**). The hub then assigns each email to an event queue (**4**). A hub can have multiple event queues (three in this example), configurable by e-mail size. Registry settings on the host machine (**1**) define the size band for each queue.

Each queue can be served by multiple policy engines (**5**). Registry settings on the host machine (**1**) specify the policy engines allocated to each queue. To minimize processing times, if a queue is empty then idle policy engines assigned to that queue are also permitted to poach messages from other queues (**6**), but only from queues with a smaller maximum size limit.

After a policy engine has successfully processed an email, the e-mail is passed back to the completion queue (**7**) on the hub before being finally returned to the source application (**2a** or **2b**).

More information:

Hub Event Queues (see page 95)

Hub Event Queues

For each hub, there is always a default queue that can hold messages of unlimited size. To allow fast-track processing of small messages, you need to create one or more additional, size-restricted queues. You do this by editing the registry.

Queue Settings

For each additional queue, you can specify:

- **The maximum size of queued messages**

  If a message exceeds this maximum size limit, the hub automatically assigns the message to the next appropriate queue.

  For example, two additional queues are defined: Small (for messages up to 10 KB) and Medium (up to 100 KB). If a 15 KB message arrives at the hub, it is immediately assigned to the Medium queue; conversely, if a 2 MB message arrives, it is immediately assigned to the default queue, which handles messages of unlimited size.

- **Dedicated policy engines available to process queued messages**

  You can specify separate lists of policy engines available for processing each queue. However, to minimize processing times, if a queue is empty then any idle policy engines assigned to that queue are also permitted to poach messages from other queues (but only from queues with a smaller maximum size limit).

Note: The default queue is not represented in the registry by a dedicated registry key and has no maximum size limit for queued messages; by contrast, any additional queues **are** represented by a dedicated registry key.

Monitoring the Queue Status

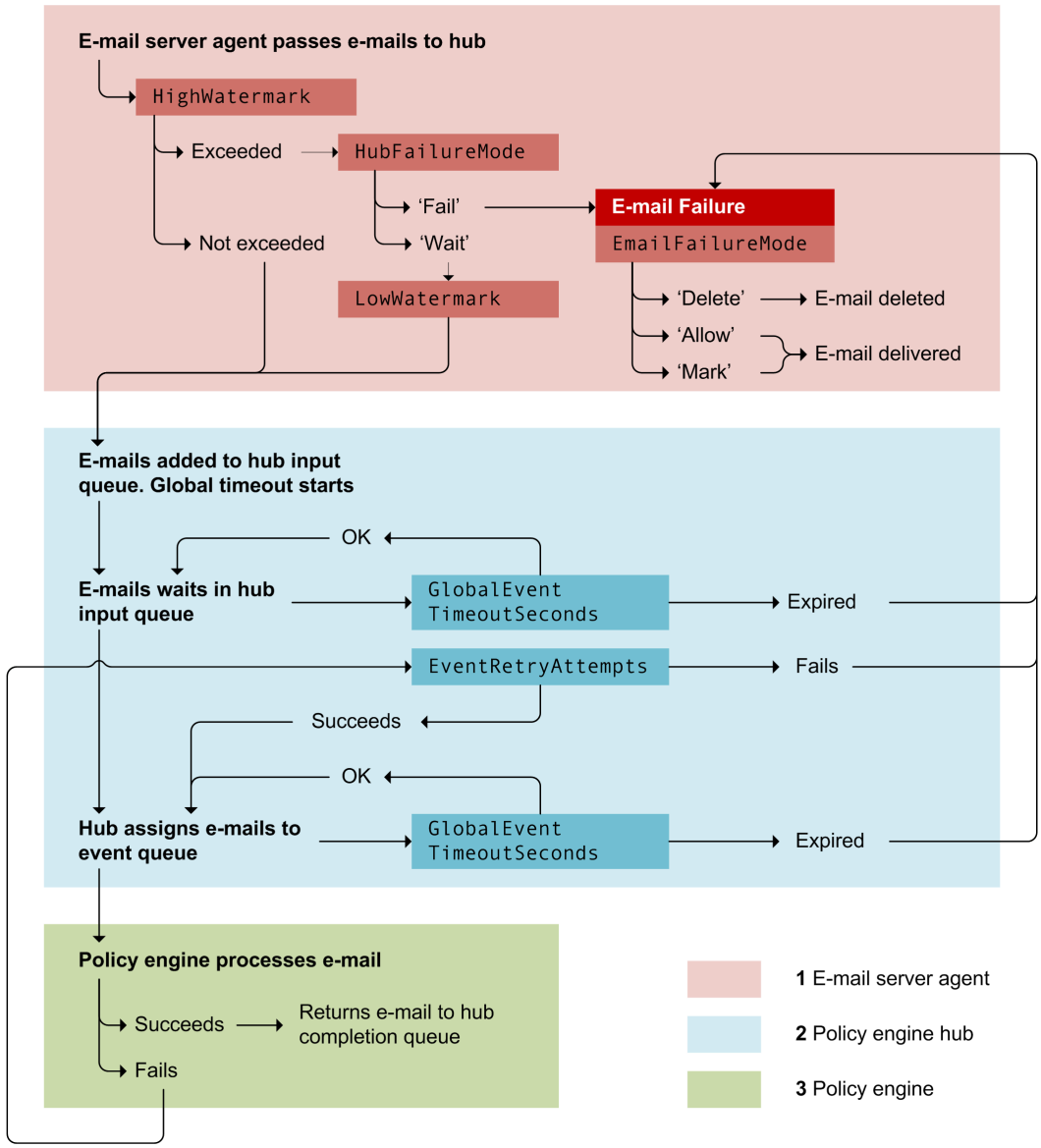
To monitor the status of individual event queues, check the relevant counters in the CA DataMinder Hub Queues performance object.

More information:

Modify the Hub Registry Values (see page 100)

Registry Flow Chart: Email Processing on the Hub

The diagram below shows how the crucial registry values for the policy engine hub and the Exchange or Domino server agent operate in a strict sequence, with the event finally passing to a policy engine.



More information:

Modify the Hub Registry Values (see page 100)
Email Server Agent Registry Values (see page 22)

Deploy the Policy Engine Hub

Deploying a policy engine hub requires the following tasks:

To deploy the policy engine hub

1. Deploy your policy engines.
2. Set up the event source.

   For example, to integrate with Exchange Server, you must deploy the CA DataMinder Exchange server agent. See the relevant Integration guidies for installation instructions.
3. Install the PE hub.

   A hub is installed automatically when you install a CA DataMinder server agent or Import Policy.
4. Configure the hub.
   a. Assign a security privilege to the PE domain user.
   b. Edit the hub registry values.

More information:

Configure the Policy Engine Hub (see page 98)
Install the Policy Engine Hub (see page 98)

Hub Host Machine Requirements

The following are requirements for the machine hosting the policy engine hub:

PE domain user must be local administrator

The PE domain user must be a member of the local Administrators group on the machine hosting the policy engine hub. Confirm that this is so before installing the policy engine hub.

More information:

Install an Exchange, Domino or IIS SMTP Agent (see page 17)

Install the Policy Engine Hub

A policy engine hub is installed automatically when you install any of the following CA DataMinder agents. See the relevant chapters for installation instructions.

- **Email Server Agents**

  Domino Server Agent

  Exchange Server Agent

  IIS SMTP Agent

- **Archive Agents**

  EMC SourceOne

  Symantec Enterprise Vault

- **Other**

  ICAP Agent

  External Agent API

In addition, you can optionally install a remote Policy Engine Connector (a type of hub) when you install the File Scanning Agent.

Configure the Policy Engine Hub

After installing the policy engine hub, you must:

- Assign the 'Log on as a batch job' security privilege to the PE domain user on the host machine for the policy engine hub.
- Configure the policy engine hub by modifying the associated registry values on the host machine.

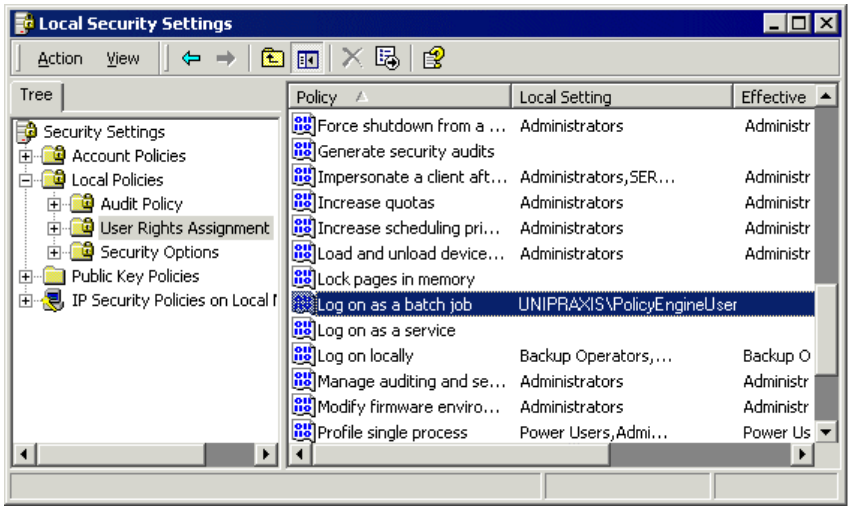
More information:

Registry Flow Chart: Email Processing on the Hub (see page 96)

Assign Security Privilege to the PE Domain User

The PE domain user requires the ‘Log on as a batch job’ security privilege. This permits policy engines on remote machines to access the policy engine hub. To assign this privilege:

1. Ensure that you are logged on with local administrator rights on the host machine for the policy engine hub.
2. On the host machine, open the Local Security Policy applet or, if this machine is a domain controller, open the Domain Controller Security Policy applet. Both applets are available in Administrative Tools.
3. Expand the Local Policies branch and select User Rights Assignment. This security area determines which users have logon privileges on the local computer.
4. Assign the ‘Log on as a batch job’ privilege to the PE domain user.



Local Security Policy applet

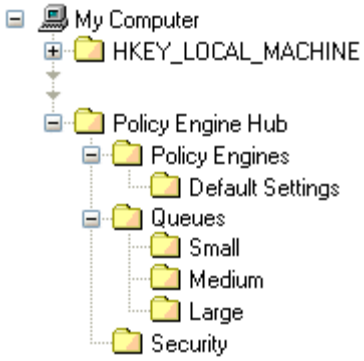
In this applet, the left pane shows the Local Policies branch and the User Rights Assignment node. The 'Log on a batch job' policy, or privilege, is shown in the right pane.

Modify the Hub Registry Values

To configure the policy engine hub, you need to modify values in the following registry key.

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\CA
DataMinder\CurrentVersion\Policy Engine Hub
```

Changes to the Policy Engine Hubs key take immediate effect. Below this registry key there are various subkeys. The key structure is shown below:



Policy engine hub registry keys: Registry values in the Policy Engine Hub key and its subkeys are described in the following sections.

Important! If the hub is deployed with the Exchange 2007 or 2010 server agent on a 64-bit machine, the registry key is slightly different.

More information:

Policy Engine Hub Registry Values (see page 101)

Policy Engine Hub Registry Values

The table below lists the available registry values for policy engine hubs.

- Policy Engine Hub key (see page 102)
  EventLoggingLevel
  EventRetryAttempts
  GlobalEventTimeoutSeconds
  HighWaterMarkMB
  HighWaterMarkEventCount
  LogFilePath
  LogMaxNumFiles
  LogMaxSizeBytes
  LowWaterMarkMB
  LowWaterMarkEventCount
  NoPEFailTimeoutSeconds
  OperationalLoggingLevel
  PECallTimeoutMilliseconds
- DefaultSettings subkey (see page 106)
  HeartbeatPeriodMilliseconds
  MetricsPeriodMilliseconds
  ReconnectTimeoutSeconds
- Queues key (see page 108)
  ActivePolicyEngines
  AdditionalQueues
  StandbyPolicyEngines
- <Queue name> subkey (see page 109)
  ActivePolicyEngines
  MaxSizeBytes
  StandbyPolicyEngines
- Security key (see page 109)
  NTNetworkDomain
  NTNetworkUser

More information:

Modify the Hub Registry Values (see page 100)

Policy Engine Hub Key

The Policy Engine Hub registry key contains the following registry values:

EventLoggingLevel

Type: REG_DWORD

Data: Defaults to 2. This determines the level of logging for message processing. For example, you can configure the hub to only log errors or warning system messages.

Log entries are written to the wgnphub_<date>.log file, where <date> is the date and time when the log file was created; the file location is set by the LogFilePath registry value. The supported logging levels are:

1 - Errors only

2 - Errors and warnings

3 - Errors and warnings, plus informational and status messages

Note: Setting EventLoggingLevel=3 will cause the log file to grow extremely rapidly. This level of logging is provided for testing purposes only.

EventRetryAttempts

Type: REG_DWORD

Data: Defaults to 4. Determines how many times the hub attempts to pass an email to a policy engine before it is flagged as an 'email failure' and passed back to the Exchange or Domino server agent.

This **only** applies to email failures caused by a problem with the policy engine (such as a host machine crash) or with the email itself (that is, some unexpected condition that prevents the policy engine from analyzing the email).

It does not apply to email failures resulting from the time-out GlobalEventTimeoutSeconds expiring or because the HighWaterMarkEventCount or HighWaterMarkMB thresholds have been exceeded.

How the email server agent handles 'email failures' depends on its EMailFailureMode registry value.

Note: For Import Policy jobs, we recommend a value of 0, to stop the policy engine retrying failed events.

GlobalEventTimeoutSeconds

Type: REG_DWORD

Data: Defaults to 300. A time-out (in seconds) that specifies how long an email can stay in the event queue on the hub or policy engine before it is flagged as an 'email failure' and passed back to the source component (typically the Exchange server agent or, for Import Policy jobs, Event Import).

How the Exchange server agent handles 'email failures' depends on its EMailFailureMode registry value (see link above). For Event Import, the handling of import failures depends on the type of import operation.

Note: If Import Policy is installed, the default for this value changes to 21,600 seconds (6 hours), so that events are less likely to timeout.

HighWaterMarkMB

Type: REG_DWORD

Data: Defaults to 400. The maximum amount of memory (in MB) that can be allocated to the various hub event queues. If any queue lengthens and causes the allocated memory to rise above this maximum level, the hub temporarily suspends normal operations until the queue shortens (see LowWaterMarkMB).

While normal operations are suspended, the hub's behavior depends on the HubFailureMode registry value. Either the hub delays new emails from the email server agent, or it returns them to the server agent as 'email failures'.

How the server agent handles 'email failures' depends on its EMailFailureMode registry value (see link above).

Note: Note the following:

- Memory-based throttling operates in parallel with event-based throttling (for details, see HighWaterMarkEventCount). If either threshold is exceeded, hub operations are suspended.
- For Import Policy jobs, we recommend a low value (for example, 40MB) to ensure a steady stream of events.

HighWaterMarkEventCount

Type: REG_DWORD

Data: Defaults to 400. The maximum total number of events that can be allocated to the various hub event queues.

If any queue lengthens and causes the event count to rise above this maximum level, the hub temporarily suspends normal operations until the queue shortens (see LowWaterMarkEventCount below).

While normal operations are suspended, the hub's behavior depends on the HubFailureMode. Either the hub delays new emails from the Exchange or Domino server agent, or it returns them to the server agent as 'email failures' (whose handling is dependent on the EMailFailureMode; see link above)

Note: Note the following:

- Event-based throttling operates in parallel with memory-based throttling (see HighWaterMarkMB). If either threshold is exceeded, hub operations are suspended.
- For Import Policy jobs, we recommend a low value (for example, 40) to ensure a steady stream of events.

LogFilePath

Type: REG_SZ

Data: Defaults to empty. This specifies the folder you want to write log files to. The PE domain user must have write access to the specified folder.

If the path is not defined, the log file is saved in the default location.

In the current CA DataMinder release, log files are typically saved in CA's \data\log subfolder. On 32-bit machines, find this subfolder in the Windows All Users profile. On 64-bit machines, find this subfolder below the \ProgramData folder.

LogMaxNumFiles

Type: REG_DWORD

Data: Defaults to 10. This specifies the maximum number of log files. When the maximum number of log files exists and the maximum size of the latest is reached (see below), the oldest log file is deleted to enable a new one to be created.

LogMaxSizeBytes

Type: REG_SZ

Data: Defaults to 1,000,000. This specifies the maximum size (in bytes) for each log file. When the current log file reaches its maximum size, the policy engine hub creates a new log file. Log entries are written to a wgnphub_<date>.log file—for details see EventLoggingLevel above.

LowWaterMarkMB

Type: REG_DWORD

Data: Defaults to 200. The total amount of memory allocated to the various hub queues (in MB) that triggers a resumption in hub operations.

If normal hub operations are suspended because the HighWaterMarkMB has been exceeded (see above), they only resume when the queues shorten and allocated memory falls back below the LowWaterMarkMB amount.

Note: For Import Policy jobs, we recommend a low value (for example, 20 MB) to ensure a steady stream of events.

LowWaterMarkEventCount

Type: REG_DWORD

Data: Defaults to 300. The total number of events allocated to the various hub queues that triggers a resumption in hub operations.

If normal hub operations are suspended because the HighWaterMarkEventCount has been exceeded (see above), they only resume when the queues shorten and the event count falls back below the LowWaterMarkEventCount amount.

Note: For Import Policy jobs, we recommend a low value (for example, 20) to ensure a steady stream of events.

NoPEFailTimeoutSeconds

Type: REG_DWORD

Data: Defaults to 60. This specifies how long (in seconds) the hub waits after it detects there is no active policy engine available for a specific queue before it times out events in that queue (that is, flags them as email failures).

When this timeout expires, all events in the queue are immediately flagged as email failures. This overrides the GlobalEventTimeoutSeconds (see above).

How the Exchange or Domino server agent handles 'email failures' depends on the EMailFailureMode; see link above)

Note: *We recommend that you do not change the default timeout of 60 seconds.*

OperationalLoggingLevel

Type: REG_DWORD

Data: Defaults to 3. This determines the level of logging for hub operations. For example, typical log entries cover hub installation, creating or deleting queues, the failure or suspension of policy engines, and so on.

Log entries are written to the wgnphub_<date>.log file, where <date> is the date and time when the log file was created; file name and location details and supported logging levels are the same as for EventLoggingLevel (see above).

PECallTimeoutMilliseconds

Type: REG_DWORD

Data: Defaults to 10000. A time-out (in milliseconds) that specifies how long the hub will wait to connect to a policy engine for configuration purposes before it cancels the call and assumes that the policy engine is currently unavailable.

(More information:

Email Server Agent Registry Values (see page 22)
Uninstall Policy Engine Hubs (see page 114)

Policy Engines Subkey

Below the Policy Engine Hub registry subkey (see the previous section), there is a Policy Engines subkey. This subkey contains no values; instead, it contains the DefaultSettings subkey and, optionally, a <Machine name> subkey.

More information:

DefaultSettings Subkey (see page 106)
<Machine name> Subkey (see page 107)

DefaultSettings Subkey

Below the Policy Engines registry subkey (see the previous section), there is a DefaultSettings subkey. Values in this subkey define the default configuration for all policy engines.

HeartbeatPeriodMilliseconds

Type: REG_DWORD

Data: Defaults to 40,000. This specifies how often (in milliseconds) the policy engine sends a heartbeat signal to the policy engine hub. If the hub does not receive three successive heartbeat signals, it infers there is a problem with the policy engine.

MetricsPeriodMilliseconds

Type: REG_DWORD

Data: Defaults to 40,000. This specifies how often (in milliseconds) the policy engine returns metrics to the policy engine hub. This value must be an integer multiple of HeartbeatPeriodMilliseconds.

ReconnectTimeoutSeconds

Type: REG_DWORD

Data: Defaults to 600. If a policy engine does not restart immediately when the hub tries to connect to it, this value specifies how long (in seconds) the hub waits between subsequent reconnection attempts.

Note: This timeout is only applicable if the hub fails in its initial attempts after startup to connect to a policy engine (for example, because the policy engine host machine is switched off).

<Machine name> Subkey

To override the default policy engine configuration, you can create a <Machine> subkey below the Policy Engines registry subkey. Note that <Machine> is the host machine for the policy engine you want to customize. The <Machine> subkey can contain customized versions of any registry value in the DefaultSettings subkey.

More information:

DefaultSettings Subkey (see page 106)

Queues Key

Below the Policy Engine Hub registry key, is the Queues subkey. It contains the following registry values, plus various subkeys, one for each message queue supported by the hub.

ActivePolicyEngines

Type: REG_SZ

Data: Defaults to <localhost>. This specifies a comma-separated list of names or IP addresses of machines hosting policy engines available to the *default queue*.

AdditionalQueues

Type: REG_SZ

Data: Defaults to null. This specifies a comma-separated list of additional message queues available to the policy engine hub. These queues are in addition to a default queue, which is always available. The example registry architecture diagram shows three additional queues: Small, Medium and Large.

Note: If you edit this registry value while the hub service is running, CA DataMinder automatically creates new registry subkeys for each additional queue. However, if you want to configure your hub in advance, you must manually create a subkey for each additional queue, and you must also add and configure the necessary registry values to this new subkey.

StandbyPolicyEngines

Type: REG_SZ

Data: Specifies a comma-separated list of names or IP addresses of machines, available to the default queue, and which can be used by the hub if an 'active' policy engine is unavailable.

More information:

Policy Engine Hub Architecture (see page 93)

<Queue name> Subkey

In the example registry architecture diagram, the hub supports three additional queues: Small, Medium and Large. There is a separate subkey for each of these queues. These queues are in addition to a default queue, which is always available and does not have its own registry subkey.

You must create these additional subkeys manually or, if the hub service is running when you edit the AdditionalQueues registry value, the subkey is created automatically. Each of these manually added subkeys contains the following registry values.

ActivePolicyEngines

Type: REG_SZ

Data: Defaults to <localhost>. This specifies a comma-separated list of names or IP addresses of machines hosting policy engines available to the *specified queue*.

StandbyPolicyEngines

Type: REG_SZ

Data: Specifies a comma-separated list of names or IP addresses of machines, available to the *specified queue*, and which can be used by the hub if an 'active' policy engine is unavailable.

MaxSizeBytes

Type: REG_DWORD

Data: Defaults to zero. This specifies the maximum size (in bytes) of message events that can be processed by the specified queue. If a message is too large for this queue, it is assigned to the next size queue. You must change the default value of MaxSizeBytes. If it remains set to zero, this queue can never process any messages!

More information:

Policy Engine Hub Architecture (see page 93)

Security Key

Below the Policy Engine Hub registry key, there is a Security subkey. You do not need to modify the values in this subkey because they are managed by the policy engine hub. But for reference, the values are:

NTNetworkDomain

Type: REG_SZ

Data: Domain name. Created automatically when you configure the PE domain user. *Do not modify this value directly.*

NTNetworkUser

Type: REG_SZ

Data: User name. This value is created automatically when you configure the PE domain user. *Do not modify this value directly.*

Hub Maintenance

If you need to shut down the policy engine hub (for example, while you upgrade the Exchange or Domino server agent), you must follow the recommended procedure to ensure that no emails are inadvertently deleted or transmitted without being monitored by CA DataMinder.

Stopping the Policy Engine Hub

1. You must suspend normal Exchange, IIS SMTP, or Domino operations before you stop the policy engine hub service. There are several ways to do this for Exchange 2003 and IIS SMTP, but we recommend that you stop Internet Information Services (IIS). This is because you cannot upgrade the email server agent .DLL file while IIS is running. For Exchange 2007 and 2010 we recommend that you stop the Microsoft Exchange Transport service.
2. Stop the CA DataMinder Policy Engine Hub service. You can now perform any necessary maintenance or upgrades on the host machine.

Restarting the Policy Engine Hub

You must restart the services in the reverse order to which they were stopped. That is, restart the CA DataMinder Policy Engine Hub service, then restart IIS (or the Microsoft Exchange Transport service).

Consequences If You Stop the Hub Before IIS

If you stop the policy engine hub before stopping IIS (when applicable), there is a risk that emails may be deleted or transmitted without being monitored by CA DataMinder, or that emails may be imported twice. These consequences can arise if the Exchange or Domino server agent passes emails to the hub while it is shutting down.

When the email server agent receives no response from the hub, it infers there has been an email failure and uses the EMailFailureMode registry value to determine how to handle the email: this value can be set to Delete, Allow, or Mark (see EMailFailureMode).

Alternatively, the timing of the hub shutdown may be such that an email is sent to a policy engine for processing immediately before the hub shuts down. The policy engine successfully processes the email but is unable to notify the hub. Consequently, the email is resubmitted to a policy engine when the hub restarts.

Monitor Policy Engine Hub Activity

There are various sources of diagnostic information when monitoring policy engine hubs. These are performance counters, log files and diagnostic files.

PE Hub Log Files

Note: If the hub is deployed with the Exchange 2007 or 2010 server agent on a 64-bit machine, note that the server agent and hub log files may be in different locations.

Policy Engine Hub

The policy engine hub service writes entries to the log file, wgnphub.log. These detail progress as each email is processed. This log file is in the same folder as the hub executable, wgnphub.exe, typically installed to the \System subfolder in the CA DataMinder installation folder on the Exchange or Domino server.

The hub writes entries to this log file using the PE domain user account. By default (on NTFS file systems), security for wgnphub.log has been configured to give the PE domain user Read and Write access to this file.

You configure the policy engine hub log files by editing the relevant registry values..

Exchange, IIS SMTP, and Domino Server Agents

For details about the Exchange, IIS SMTP, and Domino server agent log files, see Log files for email server agents .

More information:

Policy Engine Hub Registry Values (see page 101)
Log Files for Email Server Agents (see page 54)

Hub Performance Counters

The policy engine hub includes three performance objects that are useful when diagnosing hub problems. In the Performance applet (accessible from Administrative Tools), you can add a range of useful performance objects. For each performance object, you can specify which counters and, where relevant, instances you want to view.

Note: On a 64-bit system, all CA DataMinder performance counters, including counters for a 32-bit policy engine hub, are supported in the default 64-bit version of the Performance applet. See the following section for details.

Policy Engine Performance Object

For policy engine performance object details, see 'Monitor Policy Engines' in the *Platform Deployment Guide*.

Hub Performance Objects

For policy engine hubs, the following performance objects are available:

CA DataMinder Hub

Available only as a single instance. This contains counters for the policy engine hub itself. For example, you can see the number of active, standby and connected policy engines, the number of pending events in the hub input queue, the number of event failures, and the total memory allocated to events in the queue.

CA DataMinder Hub Connections

Multiple instances available; one per policy engine. This performance object contains counters for hub connections to individual policy engines. To view counters for a connection to a specific policy engine, select its corresponding instance.

Available counters show statistics such as the number and rate at which events are being passed to the policy engine and the internal state of the policy engine, for example, ‘inactive’, ‘processing’ and ‘dead’ (if you encounter a problem with a policy engine, Technical Support may ask for details of its internal state).

CA DataMinder Hub Queues

Multiple instances available; one per event queue on the hub. For each instance, this performance object contains counters for individual policy engines. To view counters for a specific queue, select its corresponding instance.

Available counters show statistics such as the queue size band (in bytes), the number of active and standby policy engines available to process the queue, and the number of items assigned to the queue.

Performance Counters Now Supported in 64-bit Perfmon.exe

On 64-bit systems, all CA DataMinder performance counters are now supported in the default 64-bit version of the Performance applet (perfmon.exe).

Previously on 64-bit systems, most CA DataMinder performance counters were only supported in a 32-bit version of the Performance applet. This anomaly particularly affected performance counters for the policy engine hub when the hub was installed on a 64-bit Exchange 2007 or 2010 server.

If you previously used the 32-bit Performance applet to monitor CA DataMinder components on a 64-bit system, you must switch to the 64-bit Performance applet after upgrading to CA DataMinder 14.1.

Background

On a 64-bit Windows operating system, there are two versions of perfmon.exe:

- A 64-bit version is available in the \Windows\System32 folder. This is the main System folder for the 64-bit operating system.

  Why 'System32'? The folder name, an apparent misnomer, is a legacy of the folder naming scheme in earlier Windows operating systems.

- A 32-bit version is available in the \Windows\SysWOW64 folder.

  Why ‘WOW64’? On a 64-bit Windows operating system, there is an emulation of a 32-bit operating system called ‘Windows on Windows 64’, or WOW64.

Uninstall Policy Engine Hubs

Important! If a policy engine hub is installed on the same computer as an Exchange, Domino, or Enterprise Vault server agent, you must uninstall the server agent before uninstalling the policy engine hub.

To uninstall a policy engine hub, you must uninstall its associated server agent; the hub is then uninstalled automatically. Use Add/Remove Programs to manually uninstall the Exchange or Domino server agents. This applet is part of the Control Panel.

1. In Add/Remove Programs, select CA DataMinder Integration Agents and click Change.
2. When the wizard starts, go to the Program Maintenance screen and choose Modify.

   Note: If you choose Remove, this removes all CA DataMinder components, not just the Exchange or Domino server agents.
3. In the Custom Setup screen, choose the Exchange Server Agent or Domino Server Agent, as required.
4. In the final wizard screen, click Install to begin the uninstallation.

IIS Restarts When Uninstalling Exchange Server Agent or IIS SMTP Agent

When uninstalling the Exchange server agent or IIS SMTP agent, the wizard stops Internet Information Services (IIS) before uninstalling the server agent and hub components. It then restarts IIS automatically when the uninstall is complete.

Note: IIS is installed automatically as part of an Exchange Server installation.

Chapter 6: Deployment Considerations

When you deploy the Exchange or Domino server agents, you need to consider the following issues.

This section contains the following topics:

Prevent Repeat Processing by Server Agents in Multiple Domains

You can use the EnterpriseDNSList and SmtpDNSHostName registry values to ensure that e-mails are not reprocessed needlessly by consecutive Exchange server agents. You can also use them to enforce selective repeat processing, for example, to configure a logical e-mail enterprise that is not constrained by its physical topology. See the following examples.

Example 1: Prevent Repeat Processing

Unipraxis Corporation has Exchange Servers in multiple domains, and each one hosts an Exchange server agent. On each host machine:

- SmtpDNSHostName is set to UNIPRAXIS.COM.
- EnterpriseDNSList includes UNIPRAXIS.COM in its domain list.

Now, when any Exchange server agent receives an e-mail tagged as coming from UNIPRAXIS.COM, it knows that policy has already been applied and so does not reprocess the email.

Note: If the Unipraxis Exchange servers are all in the UNIPRAXIS.COM domain, you do not need to edit these registry values to prevent repeat processing. By default, each Exchange server agent infers that emails arriving from this domain have already been processed and will exclude them from processing.

Example 2: Selective Repeat Processing

Unipraxis Corporation have a single worldwide domain, UNIPRAXIS.COM, but a CMS in the US and Europe. They want emails sent between US offices to be processed once, but emails sent to a European office to be processed a second time, and European policy applied.

On all their US Exchange servers, SmtpDNSHostName and EnterpriseDNSList are set to UX-US.COM. On all their European Exchange servers, these registry values are set to UX-EUROPE.COM. Specifically, the US domain is not recognized by the Exchange server agents in Europe, and the European domain not recognized by the US server agents.

This ensures that e-mails sent between US offices, or between European offices, are processed once only, but emails sent between the US and Europe are reprocessed on arrival, and a second policy applied.

More information

Email Server Agent Registry Values (see page 22)

Using Email Client Agents and Email Server Agents Together

By default, Exchange or Domino server agents do not reprocess emails that have already been processed by an Outlook or Notes client agent.

To enable the Quarantine feature in CA DataMinder installations that use both the Outlook client agent and Exchange server agent, you must explicitly configure the Exchange server agent to reprocess (and apply quarantine actions to) emails already processed by a client agent. To do this, you must edit the EnableIntegration registry value on the Exchange server.

Important! If you are using Exchange Server 2007 or 2010, you must set the UpdateConfig registry value to 1 for changes to this value to take effect.

Note: The Domino server agent does not currently support the CA DataMinder quarantine feature. The Milter MTA agent does, however, support the quarantine feature.

More information:

Email Server Agent Registry Values (see page 22)

Do Not Release 'dead' Messages in Domino Administrator

Domino administrators need to be aware that messages waiting to be processed by the Domino server agent have their routing state marked as 'dead'. Consequently, they are temporarily listed in the mail.box mailbox in Domino Administrator. If you open one of these messages, the failure reason is listed as 'Held for CA DataMinder'.

Eventually, when the message has been processed by a policy engine and returned to the Domino server agent, the message will either be allowed (that is, delivered as normal) or deleted. In either case, the message is automatically removed from the mail.box mailbox in Domino Administrator.

Do not manually release these messages; if you do, the message will be sent on to its intended recipients without being processed by the Domino server agent and a CA DataMinder policy engine.

This is particularly important if the hub cannot connect to a policy engine for any reason. Note that eventually, if the connection failure persists, any dead emails will automatically flagged as 'deleted' or 'allowed', depending on the EMailFailureMode registry value.

More information:

Email Server Agent Registry Values (see page 22)

Integration with an Exchange Server Cluster

The CA DataMinder Exchange server agent can integrate with an Exchange cluster. The deployment procedure is simple, and requires only that the server agent and policy agent hub are installed on each cluster node.

Note: CA DataMinder integration with Exchange clusters has been tested with Exchange Server 2003 on a two node cluster.

1. In a clustered Exchange environment, install the CA DataMinder Exchange server agent and policy engine hub on the primary node. The installation and configuration procedures are exactly the same as for a standard deployment.
2. Now install the Exchange server agent and policy engine hub on the secondary (and any additional) nodes. The server agent and hub **must** be configured exactly the same as on the primary node!

   Specifically, the hubs on each cluster node must use the same credentials to access remote policy engines (set with the wgnphub -SetCredentials command) and have identical registry values. For example, each hub uses the same pool of active and standby policy engines. Similarly, each Exchange server agent must have identical registry values.
3. Ensure that each hub service is correctly configured and running.
4. This completes the deployment. Now if the primary Exchange node fails over, email processing switches automatically to the Exchange server agent and policy engine hub on the secondary cluster node. In effect, CA DataMinder is now integrated with a single Exchange virtual server.

More information:

How to Deploy the Exchange, Domino or IIS SMTP Agent (see page 15)

Configure All Exchange Server Agents

For Exchange Server 2007 and 2010 only.

You must ensure that the configuration for **all** Exchange 2007 or 2010 server agents on Hub Transport Servers in an Active Directory Site is identical. Exceptions to this rule are the following registry values, which may differ for diagnostic purposes:

- CreateEML
- CreateEVF
- DiagnosticFolder
- LogLevel
- LogMaxNumFiles
- LogMaxSizeBytes
- NotificationFromAddress

After you are satisfied with this configuration, you must set the UpdateConfig registry value to 1 on **each** Hub Transport Server for any changes to take effect.

More information:

Email Server Agent Registry Values (see page 22)

Known Issues

Failure to Generate Email Events

If the Exchange server agent fails to generate email events, the following questions can help you to diagnose the problem.

Has a Policy Engine Been Activated?

Use Process Explorer (from www.sysinternals.com) to ensure that a wgnpesv.exe service is running on the policy engine host machine and using the correct user account. If it is not:

- Has the policy engine service been set to run as the correct named user?
- Have the same credentials been assigned to the policy engine hub?
- Have the policy engines been defined and configured correctly in the hub registry?
- Does the NT System or Application event log on either machine contain relevant information?

Check the policy engine hub log file.

Has a Policy Engine Stopped Working?

To force the policy engine hub to disconnect and reconnect to a policy engine, remove the policy engine from the ActivePolicyEngines registry value then add it again. This is a policy engine hub registry value.

Note: For the Exchange server agent, if you need to restart the policy engine hub, you must first stop the Internet Services.

Is Email Address Mapping Set Up Correctly?

- Check the machine policy settings on the policy engine host machine.
- Check that the UserSpecificAddrPattern registry value is correctly configured with a pattern match for the email addresses you want to detect. This is a policy engine hub registry value.

Is wgnemno.dll Registered as an Add-in?

(Domino sever agents only) Confirm that the following line has been added to notes.ini on the Domino host server. Find this file in the same folder as the Domino executables, typically \Lotus\Domino.

```
EXTMGR_ADDINS=wgnemno.dll
```

If this line is not present, manually add this line and restart Domino.

Unable to Expand Distribution Lists with Hidden Membership

(Exchange sever agents only) The fix for enabling a policy engine to expand hidden distribution lists depends on which version of Exchange Server you are using.

Fix for Exchange Server 2003 or Later

If a distribution list in Exchange's Global Address list has been configured to hide list members, policy engines will be unable to expand these distribution lists to identify and apply policy triggers to individual recipients.

If you want policy engines to expand these distribution lists, you must ensure that the policy engine service account belongs to a group whose members have the necessary permissions to expand 'hidden membership' distribution lists.

Multiple Notifications in Response to a Single Email

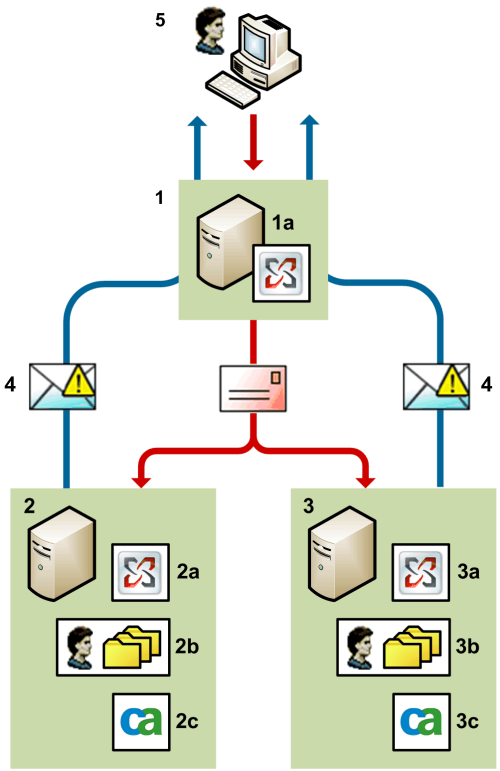
When an email activates a control trigger, the sender of the email can sometimes receive multiple warning or notification messages. This happens when CA DataMinder applies policy to multiple copies of the original email.

This can occur if an email is sent, via an email server with no server agent installed. In this situation, the following scenarios can result in multiple notification messages being sent:

- The recipients' mailboxes are hosted on multiple email servers. Each of these email servers has a server agent installed. Policy is applied to the email each time it reaches the server agent on one of these subsequent email servers. This example is shown below.
- Two or more recipients have mailboxes hosted on the same email server. The 'sending' email server sends copies of the email to each recipient's mailbox. The server agent on the email server hosting these mailboxes applies policy to each recipient's copy of the email.

The two scenarios above are the most likely cause of multiple notification messages. But other configurations can also cause multiple notifications. For example, multiple copies of the original email may simply be routed through an email server hosting the server agent. Likewise, a non-standard journaling setup may result in the server agent failing to recognize a journaled email and applying policy to that email.

To avoid any of the above scenarios, we recommend that you install server agents on all email servers in the CA DataMinder enterprise.



Multiple notification messages example

1 A user sends an email and it transits through the email server (**1a**), undetected. The recipients' mailboxes (**2b** and **3b**) are on two separate email servers (**2a** and **3a**).

2 and **3** An instance of the email arrives at each email server. Each instance is detected by an email server agent (**2c** and **3c**) and triggers a warning notification.

4 Each email server agent then sends a warning message to the sender.

5 The user receives multiple notification messages in response to a single email.

Index

P

Q

R

S

U

W

X