CA DataMinder

Archive Integration Guide

Release 14.5

2nd Edition

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.

Contents

Chapter 1: About this Guide

This guide focuses on how captured data is ingested into CA DataMinder. In particular it describes:

- Event Import operations to import emails and files into the CMS database from email archives and file storage locations.
- Policy operations, whereby policy is applied to events as they are imported.
- The IM Import utility and how to import IM conversations from dump files.
- CA DataMinder integration with third party archive solutions.
- Policy engine hubs.

Chapter 2: Event Import

This section introduces the Event Import utility.

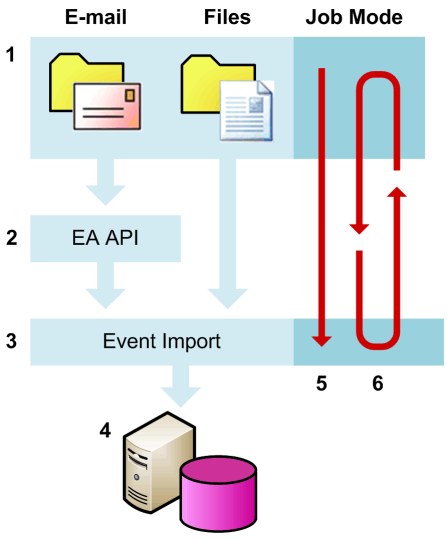
This section contains the following topics:

About Event Import

Event Import enables CA DataMinder to integrate with third-party email archives and file storage locations. In particular, Event Import can import:

- **Emails:** Typically, these are emails that have been extracted from an archive. Event Import can also import emails from:
  - Microsoft Exchange mailboxes
  - Lotus Notes NSF files
  - Archive files such as PST and MSG files
- **Files:** Event Import can also import files from designated folders. For example, it can import Microsoft Word documents or PDF reports. Typically, files are imported as part of an Import Policy operation in order to categorize, or apply smart tags to, important business documents.

Event Import automatically associates imported events with their correct 'owners'. If required, it can even create new users to 'own' imported events. CA DataMinder provides tools to allow Event Import to run in batch mode or to run continuously. The following sections explain how to install and configure Event Import and includes instructions for all Event Import parameters (these are used configure the import operation).



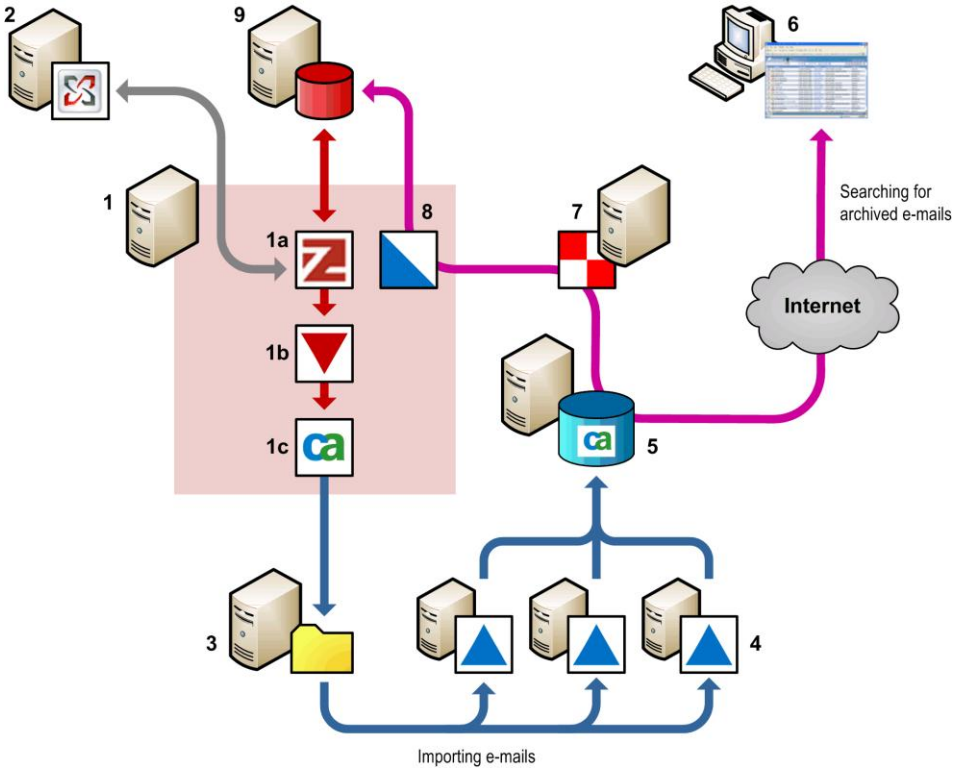
Importing email and file events

The event source (**1**), such as a folder or Exchange mailbox, is defined in a configuration file. The External Agent (**2**) receives archived emails and saves them as CA DataMinder event files. Event Import (**3**) imports the archived emails or files into the CMS (**4**). Import operations can run in batch mode (**5**) or continuously (**6**).

Importing from Archives

The principal purpose of the Event Import utility is to enable CA DataMinder to integrate with email archiving products and extract emails from an archive into your CMS. The actual message data imported from the email archive is not saved on the CMS. Instead, a record in the CMS database for each imported email references the associated entry in the archive.

The diagram below summarizes the key components and processes involved when integrating CA DataMinder with an email archive solution. This example is based on the Zantaz EAS solution. For simplicity, the diagram shows a single email archive server, feeding data into a single EVF file cache. In practice, a large organization may have many such servers feeding data into multiple caches.



Example CA DataMinder integration with email archive

1 This server hosts the email archive solution such as Zantaz EAS (**1a**). This connects to an email server such as Microsoft Exchange (**2**) and archives messages in the email store (**9**).

The email archive solution uses an indexer process such as the EAS IndexerService.exe (**1b**) which in turn passes data to the External Agent API (**1c**).

The External Agent API extracts archived emails and saves them as EVF files in a cache (**3**). This cache provides the source data for the CA DataMinder Event Import utility (**4**). This utility requires the CA DataMinder infrastructure. For very large email archives, you may need to run multiple Event Import utilities simultaneously to avoid import bottlenecks.

Each Event Import utility imports archived emails into the CMS (**5**). The actual message data is not saved on the CMS; instead, a record in the CMS database for each imported email references the associated entry in the email store (**9**).

When displaying captured emails in the iConsole (**6**), the Remote Data Manager utility (**7**) retrieves data for emails archived in the email store (**9**). In the case of EAS, these data requests are sent via Microsoft IIS (Internet Information Services) (**8**).

Identifying the Owners of Imported Events

Event Import does not assign imported emails directly to owners. Instead, it identifies 'event participants' and associates an email address with each participant. Under normal conditions, these participants are only mapped to CA DataMinder users during event searches, not while an import job is running. Of course, this mapping mechanism requires that users' email accounts are kept synchronized with their CA DataMinder accounts.

However, address mapping is used during an import job if an 'attribute-based' import filter is specified. This enables import jobs to exclude or only include emails associated with CA DataMinder users who have specific account attributes.

More information:

Synchronize Email Accounts and CA DataMinder Users (see page 19)
Filtering Email Import Operations (see page 19)

Synchronize Email Accounts and CA DataMinder Users

Important! To successfully integrate CA DataMinder with your email archive solution, good synchronization is critical between your e-mail user accounts and your CA DataMinder user accounts!

To ensure that each imported event is associated with its correct owners in the CA DataMinder user hierarchy, each e-mail processed by Event Import needs to map directly to a CA DataMinder user.

However, the continual changes to your workforce mean that new email accounts are created, redundant accounts are deleted and existing accounts are modified. You therefore need a strategy to maximize account synchronization with CA DataMinder and to minimize anomalous accounts (existing in one system but not the other). The principal way to achieve this is by making regular use of the Account Import utility—see Synchronizing users.

Emails Ignored by Event Import

Certain categories of email are ignored by Event Import. That is, they are not imported into the CMS. These include:

'Non-mail' messages

These include emails that are neither incoming nor outgoing. For example, this includes the Outlook welcome message and draft messages. They also include other mailbox items such as appointments.

Out of date emails

If an import job is set up to use the parameters EMail.EventDateFromEMail and Engine.EventRetentionPeriod, then Event Import ignores emails where the retention period has expired.

Filtered email

You can set up import jobs to only include or to exclude certain categories of email.

More information:

Engine Parameters (see page 45)
Filtering Email Import Operations (see page 19)

Filtering Email Import Operations

If required, you can filter import jobs to exclude or only include certain categories of email.

Filtering by Sender Address

If required, you can filter import jobs to exclude or only include e-mails based on the sender's email address. To do this, you use these parameters:

`EMail.SenderAddrIncludeFilter`

`EMail.SenderAddrExcludeFilter`

You typically use these parameters to only import internal emails, or to only import internal emails but exclude emails sent by specific users or groups of users.

More information:

Event Import Parameters (see page 41)

Filtering by User Attributes

If required, you can configure import jobs to exclude or only include e-mails associated with CA DataMinder users who have specific account attributes. For example, if your CA DataMinder user accounts include a 'Country' attribute, you can configure import jobs to only import e-mails sent by or to users in a specific country.

The import filter is defined in the machine policy of the Event Import host machine, where the User Filter policy setting defines a lookup expression containing one or more True or False tests that relate to a single CA DataMinder user attribute (for example, a user's team or country). Find this setting in the Email User Identification folder.

Event Import uses address mapping to associate each event participant, via an email address, with a CA DataMinder user. It then evaluates the lookup expression for that user.

- If **all** the users **fail** the test (in each case, the command evaluates to False), the email is **not** imported.

  Note: If the relevant user attribute is blank or not defined for any user, that user is deemed to have failed the test.

- If **any** user **passes** the test (the command evaluates to True), or any participant cannot be mapped to a CA DataMinder user (because no corresponding CA DataMinder user exists), the event **is** imported.

The expression syntax is:

```
<uservar><operator><text>
```

where:

<uservar>

Specifies the name of the user attribute you want to test against.

<operator>

Determines whether the value specified by <text> must be present or absent.

<text>

Represents the attribute value whose presence or absence you want to test. Values are not case-sensitive; use double-quotes if the value includes spaces.

Filter expressions leverage the existing User Attribute data lookup functionality. For full details about <uservar>, <operator> and <text>, see the Administration console online help; search the index for ‘userattr commands, data lookup’.

For example, to exclude all emails associated with users based in your organization's UK office:

1. Customize the CA DataMinder user attributes for your organization so that the ‘Country’ attribute specifies where the user is based.
2. In the machine policy for the host machine, specify this expression in the User Filter setting:

   ```
   country is not uk
   ```

3. Any user whose ‘Country’ attribute is not ‘UK’ will fail the test (the expression evaluates to False).

Support for Exchange Envelope Journaling

Event Import can import emails from Exchange servers that have envelope journaling enabled.

The envelope journaling feature in Exchange enables organizations to archive transport envelope data. This includes the actual recipient information that the transport system used to route and to deliver the email. In particular, it identifies the recipients who actually received the message, including recipients from distribution lists and blind carbon-copy (Bcc) recipients.

Unlike message-only journaling, which copies emails to be archived into a designated mailbox, envelope journaling copies into the designated mailbox an envelope message (a 'wrapper' email) that contains a journal report plus an attachment containing the original email. The journal report (the body text of the envelope message) contains the transport envelope data of the original email.

However, when an email is imported from a mailbox on an Exchange server that has envelope journaling enabled, CA DataMinder extracts the journal report and the attachment containing the original email and discards the envelope message. This process is referred to as 'de-enveloping'. CA DataMinder then creates a new event based on the original email and the actual recipient details. When a reviewer searches for this imported email in the iConsole or Data Management console, the console displays the original email.

Starting with Microsoft Exchange Server 2010, Information Rights Management (IRM) allows Microsoft Outlook and Microsoft Office Outlook Web App users to protect their messages. In the context of envelope journaling, "journal report decryption" allows Exchange administrators to save a clear-text copy of IRM-protected messages in the journal emails, along with the original, IRM-protected message.

When CA DataMinder encounters such journal emails, it extracts the clear-text copy of the IRM-protected messages.

If journal report decryption was disabled at the time an IRM-protected message was journaled, then the journal email only contains the original, IRM-protected message, and CA DataMinder can only extract the IRM-protected message.

Note: Recipients extracted from the journal report are searchable in the iConsole or Data Management console. That is, reviewers can specify a recipient when searching for emails imported from an Exchange server that has envelope journaling enabled.

Single Import Operations Only from Each Exchange Mailbox

We strongly recommend that you only run one import operation at a time from a single Exchange mailbox. This is to avoid the risk of generating duplicate imported events on the CMS. For example, if two Event Import operations (running on separate machines) are simultaneously importing emails from the 'Frank Schaeffer' mailbox, there is a risk that individual emails sent to or from Frank Schaeffer are imported twice, generating duplicate events on the CMS. There is also a lesser risk that some emails are not imported at all.

Note: You can, however, run multiple import operations simultaneously from separate Exchange mailboxes. Note also the "Messaging API and Collaboration Data Objects 1.2.1" ("MAPI client and CDO") requirement for mail box import operations

Emails Sent from Outlook 2003 in Cached Exchange Mode

E-mails sent from Outlook 2003 when using Cached Exchange mode have truncated timestamps. That is, the timestamps for these e-mail events are rounded down to the nearest minute. This can cause problems as CA DataMinder uses this timestamp to record when the event was captured.

Event Import Requirements

Note the following Event Import requirements.

Exchange and Outlook Import Requirements

Before running an import operation to import emails directly from an Exchange server or emails in .MSG or .PST files, note the following requirements.

Exchange Server

Event Import can import emails sent using Microsoft Exchange Server 2003, 2007, or 2010.

Note: Event Import is unable to import emails from journal mailboxes on Exchange 2013 servers. This limitation is caused by changes in Exchange 2013 support for MAPI. We anticipate that this limitation will be fixed in a future CA DataMinder release.

Event Import Host Machine

Install Event Import on a gateway server.

PST and MSG type operations

The Event Import host machine must be running:

Microsoft Outlook 2003, 2007, 2010, or 2013

Outlook *must* be the default email application on the host machine.

Important! You cannot use an Exchange Server Management Pack instead of Outlook on the host machine! You *must* use 32-bit Outlook as your MAPI client. You cannot use 64-bit Outlook to import .MSG or .PST files.

MSG only

To import .MSG message files saved in Outlook 2003, Outlook 2003 must be installed on the Event Import host machine. If an earlier version of Outlook is installed, the import operation fails.

PST only

If using Outlook 2003 or later, you can create two types of .PST archive file, a 'Personal Folders file' or a 'Personal Folders file (97-2002)'.

The 'Personal Folders file' is not compatible with pre-2003 versions of Microsoft Outlook. To import a 'Personal Folders file' into CA DataMinder, you must have Outlook 2003 or later on the Event Import host machine. If you have an earlier version of Outlook, the import operation fails.

EXCH type operations

Mailbox user must be in GAL

When you import emails directly from mailboxes on an Exchange server, the user associated with the source mailbox must be in the Exchange Global Address List (GAL) at the time of the import operation.

MAPI client and CDO 1.2.1

The Event Import host server requires Microsoft Exchange Server "Messaging API and Collaboration Data Objects 1.2.1".

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.

Important! Before you install the 'MAPI client and CDO 1.2.1' component on the Event Import host machine, verify that Microsoft Outlook is *not* also installed. If Outlook *is* installed, uninstall it.

More information:

Event Import Types (see page 35)

Events Abandoned by Event Import

Events can be abandoned when the importer is carrying out an import operation and:

- Event Import is on a standalone machine, the infrastructure is suspended, and then the importer is shut down, or
- Import Policy is configured to Direct or Hub mode, the import engine is requested to shut down, but the policy engine and the hub do not complete message processing in a timely fashion.

If Event Import abandons an email event that it was processing, it leaves the source data unchanged in its source directory. For example, if an event is associated with:

- An MSG file, the source MSG file is left intact in the directory it was being imported from.
- A PST file, the source PST file (including its contents) is left intact in the directory it was being imported from.

Notes Import Requirements

Before running a Notes import operation, note the following requirements.

Email Server

Event Import can import emails from Lotus Domino 7.0.2, 8, or 8.5

Event Import Host server

Install Event Import on a gateway server.

Lotus Notes

The Event Import host machine must be running:

Lotus Notes 7, 8, or 8.5

Notes User

The Notes user on the Event Import host server must have access to all Notes databases that you want to import emails from. If you intend to use the NSF.DeleteAfterImport parameter, the Notes user must have sufficient rights to delete emails after a successful import operation.

CA DataMinder needs to access the Notes user on the Event Import host server. You therefore need to set the password for this user. To avoid a security loophole, configure wgncred.exe to securely cache this password. The corresponding parameter for setting this password in import.ini is NSF.ImportPassword.

Note: When importing emails from a Notes database, be aware that you cannot import directly from the Trash and Junk Mail folders.

Importing from an encrypted Domino journal mailbox

The Notes user on the Event Import host machine must also be the user that the Domino journal is encrypted on behalf of.

The Notes user on the Event Import host machine must have rights to manage the journal mailbox. To grant these rights:

a. Log onto the Domino server and right-click the journal mailbox, mailjrn.nsf.

b. Click Access Control, Manage.

c. Set the User Type to 'Person'.

d. Set the Access to 'Manager'.

e. Click the 'Delete Documents' check box.

More information:

NSF File Parameters (see page 62)

Logon Requirements for the CMS - Event Import Requirements

When you run Event Import, you must log on to the CMS as a CA DataMinder administrator. You can do this using import parameters (the parameters to define the CMS logon account are Engine.BulkImportUserName and Engine.BulkImportUserPasswd), though typically you cache the credentials. But note the requirements for this account:

- **Management group:** This account must have a management group that encompasses all the users against which you are importing emails. If the management group is too restricted, Event Import will be unable to create new users and will fail to import emails whose owners are CA DataMinder users in groups outside of the management group.
- **Administrative privileges:** This account must have the 'Events: Allow event import' and 'Events: Allow bulk session management' privileges.

Note: For details about administrative privileges and management groups, see the Administration console online help; search the index for 'privileges'.

More information:

Continuous Import Operations (see page 31)

Logon Requirements for Event Import

Before you run Event Import, you may need to change your logon details or the logon details of the Event Import service, depending on the source location of the imported files.

Important! The logon account for the Event Import service must have the administrative rights to access the source location.

Wgnimp.exe and Batch Importing

Before you run wgnimp.exe, depending on the source location of the imported files, you may need to log on to the Event Import machine using an account with administrative rights to access the source location.

- **Importing from a shared network folder**

  You must run wgnimp.exe using a logon account that has permission to change the source folder.

- **Importing from Microsoft Exchange mailboxes**

  You must run wgnimp.exe using a logon account that has permission to access the source mailboxes.

Wgnimpsv.exe and Continuous Importing

By default, wgnimpsv.exe logs on as LocalSystem. Depending on the type of import operation, you may need to change this to a user account with administrative rights to access the source location.

- **Importing from a shared network folder**

  Wgnimpsv.exe must log on using an account that has permission to change (write to) the source folder.

- **Importing from Microsoft Exchange mailboxes**

  Wgnimpsv.exe must log on using an account that has permission to access the source mailboxes.

Important! If you change the logon account for wgnimpsv.exe from LocalSystem to a named user account, this account must be a member of the local Administrators group. This ensures that the import service has access to the registry and, if CA DataMinder is installed to its default location in the Program Files folder, that it can write progress messages to logfiles.

Installing Event Import

We recommend that you install Event Import on a gateway server. Gateways inherit the common gateway machine policy. You must confirm that the Infrastructure settings in this policy are appropriate for your network.

To install Event Import, you run the CA DataMinder server installation wizard.

To install Event Import

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 CA DataMinder Platform and then click Install.

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

4. In the server installation wizard, navigate to the Customer Setup screen.

5. In the Custom Setup screen, expand the Enterprise Server feature and choose Event Import.

   We recommend that you also install the Templates subfeature. This installs the template import configuration files. Each of these contains the minimum parameters needed for a specific type of import operation (for example, importing from Exchange mailboxes).

6. In the Server Type screen, choose a gateway server.
7. In the final wizard screen, click Install to start the file transfer.

More information:

Template Configuration Files (see page 36)

Running an Event Import Operation

For individual import operations, importing messages or files in batch mode, you run the Event Import utility, wgnimp.exe, from a command line.

To set up continuous import operations, you run the Event Import service (wgnimpsv.exe). If you run a continuous import operation using wgnimpsv.exe, note that wgnimp.exe remains available for concurrent individual import operations.

- **Import parameters:** You configure import operations with command line parameters or parameters in a configuration file.

  Event Import supports a wide range of parameters for configuring the import operation. For example, you can specify input queue thresholds to prevent performance bottlenecks. You can specify these parameters in a command line or, more commonly, in a configuration file
- **Logon requirements:** You must also ensure that the account you use to logon to the CMS has appropriate administrative privileges. Finally, some types of import operation, particularly when importing from Microsoft Exchange mailboxes, have specific logon requirements.

More information:

Wgnimpsv.exe and Continuous Importing (see page 28)

Event Import Operations

The CA DataMinder Event Import utility is a command line utility for importing e-mails or files into the CMS. It is also the second CA DataMinder component (after the External Agent API) in the process of extracting archived emails and importing them into a CMS.

Note: To optimize data storage when importing from an **email archive**, the actual message data is not saved on the CMS. Instead, a record in the CMS database for each imported email references the associated entry in the email archive.

- **Import parameters**

  Event Import supports a wide range of parameters for configuring the import operation. For example, you can specify input queue thresholds to prevent performance bottlenecks. You can specify these parameters in a command line or, more commonly, in a configuration file.

- **Running import operations**

  Event Import can be run in batch mode from a command line, using the wgnimp.exe utility. Or you can set up continuous import operations, using the Event Import service, wgnimpsv.exe.

More information:

Event Import Parameters (see page 41)
Running an Event Import Operation (see page 29)

Individual Import Operations

To import archived events in batch mode, you run wgnimp.exe in batch mode from a command line.

For ease of maintenance, we strongly recommend that you use a configuration file to specify your import parameters. You need to define your configuration file before running wgnimp.exe. Note also the logon requirements for:

- Wgnimp.exe
- The account that you use to log on to the CMS

To run an import operation, run wgnimp.exe. The command syntax is:

```
Wgnimp [options]
```

where [options] can be:

-h

Displays the usage information.

-f <file>

Specifies the name of the file containing import parameters. For ease of maintenance, we strongly recommend that you specify your parameters in a configuration file.

-p <parameter>

Specifies a single import parameter. Parameters listed on the command line take precedence over those in the configuration file. A single command can include multiple -p options.

The example below imports e-mails from MSG files. Other import parameters are contained in the file params.ini.

```
wgnimp -f params.ini -p import.type=MSG
```

Continuous Import Operations

When you install Event Import, the wgnimpsv.exe service is installed automatically. You can use this service for continuous import operations. But first you must configure the service by creating and editing a configuration file, import.ini. In detail:

1. **Create a configuration file, import.ini**

   Wgnimpsv.exe requires a configuration file containing the import parameters. You must name this configuration file 'import.ini' and save it in the same folder as wgnimpsv.exe. It can contain any of the import parameters available for wgnimp.exe.

2. **Specify the continuous import parameter:** In addition to the usual parameters, import.ini must also include a parameter to specify continual importing. Add the line:

   ```
   File.ContinuousInput=yes
   ```

   Note: Even if you have changed wgnimpsv.exe from a 'Manual' to an 'Automatic' startup service, you must still add this line to import.ini to prevent the service stopping prematurely.

3. **Securely cache the CMS logon credentials**

   When you run Event Import, you must log on to the CMS as a CA DataMinder administrator. So that you do not need to store the CMS credentials in import.ini (which would represent a security loophole), you can configure wgnimpsv.exe to securely cache the credentials itself. To do this, use the following command line syntax to open a new command window where you can enter a valid user name and password:

   ```
   wgnimpsv -setcredentials
   ```

   Note: The corresponding parameters for setting these credentials in import.ini are Engine.BulkImportUsername and Engine.BulkImportUserpasswd

   If you subsequently need to reset the CMS logon credentials, you can do so by running the following command:

   ```
   wgnimpsv -clearcredentials
   ```

4. **Configure the wgnimpsv.exe service**

   By default, the Event Import service installs as a 'Manual' startup type. You must change this to 'Automatic' startup type as soon as you have created your import.ini configuration file.

   If necessary, change the service logon properties to reflect the requirements on Wgnimpsv.exe and continuous importing.

5. **Begin the import operation**

   To do this, simply restart wgnimpsv.exe.

   Note: After wgnimpsv.exe has started, any subsequent changes to import.ini will only take effect after you restart the service.

More information:

Wgnimpsv.exe and Continuous Importing (see page 28)

Scheduling Remote CMS Import Jobs

If required, you can schedule remote CMS import jobs using the Windows Scheduled Tasks wizard. This allows you to schedule import jobs to run at regular intervals during periods of low CMS activity.

1. In the import configuration file, ensure that SQL.RunViaScheduledTask parameter is set to Yes.

   This parameter enables the import job to resume (during the next scheduled time slot) from the point at which it was stopped. But see also the Important for SQL.RunViaScheduledTask in RCI Job Setup parameters.

2. Open Scheduled Tasks in the Systems Tools folder.

3. When the wizard prompts you for the program you want to run, browse to the \Import subfolder in the CA DataMinder installation folder and choose WgnImp.exe.
4. Specify a scanning schedule as required.
5. When the wizard prompts you for a Windows user account, you must specify an account that is a member of the local Administrators group. This ensures that Event Import can access the registry and that it can write progress messages to logfiles.

   Note: Credentials for accessing the remote CMS are defined by the primary CMS import parameters.
6. In the final wizard screen, you must edit the advanced properties for the scheduled task. Specifically, you must edit the **Run** field.

   The Event Import executable, WgnImp.exe, is already included in the Run command, having been specified in step 3. You must now append the import configuration file and a named instance to this command. Note that the Run command uses the same syntax as command line scanning jobs:

   ```
   Wgnimp -f <file> -i <instance>
   ```

   where:

   -f <file>

   Specifies the name of the file containing import parameters.

   -i <instance>

   Specifies the name of an import instance. You **must** specify an instance even if you are only running one instance of Event Import.

More information:

RCI: Job Setup Parameters (see page 74)
Primary CMS Parameters (see page 77)
Event Import Parameters (see page 41)

Multiple Import Operations

You can run different types of import operations concurrently on the same host machine by running wgnimpsv.exe from a command line. Note also:

- All service instances share the same set of CMS logon credentials, so you do not need to rerun the wgnimpsv -setcredentials command.
- You must ensure all instances have been manually deleted before uninstalling Event Import, otherwise orphaned services will remain on the host machine.
- Each instance **must** import from a separate source.

Important! Import each service instance from a separate source. You cannot improve import performance by using multiple service instances to import data from a single Exchange mailbox or a single source folder. Indeed, this approach causes misleading 'import failure' messages to be written to the logfile.

The command syntax for configuring multiple service instances is shown below.

To create a non-default service instance

Each configuration file can specify a separate set of import parameters that apply only to that service instance. This example command creates a new service called wgnimpsv_lsteel, and the new service looks for a configuration file called import_lsteel.ini.

```
Wgnimpsv -instance lsteel -service
```

Note: The -instance parameter **must** come before the -service parameter

To start or stop a service instance

These example commands start and stop the new service.

```
net start wgnimpsv_lsteel
```

```
net stop wgnimpsv_lsteel
```

To delete a non-default service instance

This example command deletes the new service, provided it is not set as the default.

```
wgnimpsv -instance lsteel -unregserver
```

Note: The -instance parameter **must** come before the -unregserver parameter.

Event Import Types

When setting up an event import operation, you must specify the type of import operation.

The supported import types are listed below:

BBMAIL

Use to import Bloomberg emails from XML dump files. These are emails sent using Bloomberg terminals and archived in an XML dump file.

EVF

Use to import .EVF CA DataMinder event files. For example, you may want to import .EVF files in order to test an Import Policy configuration.

EML

Use to import .EML Internet email files. These are Microsoft representations of Internet Mail messages.

Note: EML files usually contain data for Internet emails, but note that the EML file extension also has other uses / can also be used by other email clients.

File

Use to import files from designated folders. These are typically text-based files such as Microsoft Word or PDF documents.

EXCH

Use to import events from mailboxes on a Microsoft Exchange server.

Exchange import requirements are in Exchange and Outlook Import Requirements (see page 19).

MSG

Use to import .MSG message files. These are emails saved as Microsoft Outlook message files.

MSG import requirements are in Exchange and Outlook Import Requirements (see page 19).

NSF

Use to import .NSF Lotus Notes database files.

NSF import requirements are in Notes Import Requirements (see page 26).

PST

Use to import .PST Microsoft Outlook archive files.

PST import requirements are in Exchange and Outlook Import Requirements (see page 19).

SQL

Use to import CA DataMinder events (emails and files) from a remote CMS.

More information:

Exchange and Outlook Import Requirements (see page 23)
Notes Import Requirements (see page 26)
Import Policy (see page 83)
Import Type Parameter (see page 44)

Template Configuration Files

CA DataMinder provides two types of template configuration files when you install Event Import using the server installation wizard. These files are installed to the \import\templates subfolder in the CA DataMinder installation folder.

They are used to import a range of file types into CA DataMinder, for example, CNV or PST files. Each import template file contains the minimum parameters needed for a specific type of import operation (for example, importing from Exchange mailboxes or PST files, or as part of an Import Policy job). Before running an import operation, you need to rename the configuration file to Import.ini, set the relevant parameters to the correct values, and move the template file into the same folder as wgnimp.exe (by default, this is the \import folder).

Example Import Configuration File

Each parameter begins on a new line. If a parameter value contains spaces, you do **not** need to enclose the parameter in double quotes. Below is a typical configuration file for a simple import operation from an EAS email archive.

Note that the parameters to define the CMS logon account are not included in this file (for security reasons). To add comments to a configuration file, use any of the following formats: #, // and REM. The full range of available parameters are shown in Event Import parameters.

This example specifies a continuous EVF import operation.

Parameters	Description
import.type=evf	Specifies an EVF file import operation.

Engine.StopOnError=No Engine.LogLevel=3	Non-critical importing errors are ignored. Full log details are written to the CA DataMinder system logfile.
# Specify continuous import	Adds a comment.
File.ContinuousInput=yes File.pathspec=\\W2K-UPX01\EASCache File.includesubdirs=No	Continuous importing is specified. This is essential when using the wgnimpsv.exe service. The source folder is defined as \EASCache on machine W2K-UPX01. The NBA does not write captured files to subfolders, so Event Import does not need to search subfolders.
// Specify email configuration	Adds a comment.
EMail.EventDateFromEMail=yes Email.InternalAddrPattern=unipraxis	Capture dates are based on the dates in the e-mails. When detected in an email address, the term 'unipraxis' signifies an internal email.
REM Specify import filter	Adds a comment.
Email.SenderAddrIncludeFilter=unipraxis	Only emails sent by users whose email address contains 'unipraxis' are imported.

More information:

Event Import Parameters (see page 41)

Import Failures

How Event Import handles individual import failures (events that cannot be imported and assigned to a CA DataMinder user) depends on the type of import job.

Batch Jobs Importing from Files

These are batch jobs running wgnimp.exe and importing from message files (for example, PST or MSG files). For these import jobs, each individual import failure is logged in the CA DataMinder Event Import log (see page 39).

Continuous Jobs Importing from PST or MSG Files

These are continuous jobs running wgnimpsv.exe and importing from message files (for example, PST or MSG files). For these import jobs, any file that fails to be imported is either moved into a \Failed subfolder or deleted. This prevents Event Import from repeatedly trying to import the same failed file.

Note that a PST file will be flagged as an import failure if one or more messages within it cannot be successfully extracted and assigned to an owner, even if some messages can be successfully imported.

As before, import failures are logged (see page 39). The following File.ContinuousInput and File.DeleteAfterImport parameters determine how such files are handled.

Note: If File.ContinuousInput=Yes, then File.DeleteAfterImport must also be set to Yes, or the import operation will fail to start.

Imported Events Cached If Replication Fails

If events are imported successfully, but then cannot be replicated successfully to a parent server, they are stored in the replication holding cache on the Event Import machine.

Exchange Mailbox Import Jobs

These jobs import e-mails from specified Exchange mailboxes. If an individual email fails to be imported, Event Import follows a predefined sequence of parameter tests to determine what failure handling is required:

- **When is an email is deleted?**

  This happens if an import job uses this parameter:

  ```
  MSExch.DeleteEMailAfterImport=Always
  ```

  In this case, emails are always deleted from the mailbox, whether or not the email was imported successfully.

- **When is an email is moved to an 'import failure' mailbox folder?**

  This happens if an import job uses:

  ```
  MSExch.FailedMailboxFolder
  ```

In both cases, import failures are logged (see page 39).

More information:

Exchange Server Import Parameters (see page 57)

Remote CMS Import Failures

If Event Import is unable to import an event (for example, because a policy engine fails to process an imported event), the event is saved as an EVF file to a folder specified by the SQL.FailedEVFDirectory parameter.

But if Event Import is then unable to generate an EVF file, it will instead save the failed event as an EVL file. For example, this could potentially happen if the primary CMS stores its event data in a third party object storage solution (such as EMC Centera or NetApp SnapLock) and that object store is temporarily unavailable at the time the import job runs.

EVLs are 'event link' files that point to the associated e-mail events on the primary CMS. You can open EVLs on any machine running the Data Management console to see the associated event. This allows you to identify which event failed to be imported and diagnose the reason for the import failure.

Note: For details about EVL files, see the Data Management console online help; search the index for EVL files.

More information:

RCI: Job Setup Parameters (see page 74)

Event Import log Files

During event import operations, import failure, error messages, and important system messages are logged in the CA DataMinder Event Import log. Find this logfile in CA's \data\log subfolder of the Windows All Users profile. Log file names take the format: evtimport_<date>.log, where <date> is the date and time when the log file was created.

The Engine.LogLevel parameter determines how much detail is logged.

More information:

Engine Parameters (see page 45)

Bloomberg Message Attachments

There are two types of Bloomberg-supplied archive for e-mail attachments: the first is for attachments sent using 'short format' Bloomberg messages; the second is for attachments sent using 'long format' Bloomberg Internet messaging. These archives typically follow these filename patterns:

- 'Short format' Bloomberg messaging: firm1234.061109.xml.tar.gz
- 'Long format' Bloomberg Internet messaging: f1234.inet.061109.xml.tar.gz

Before you can import Bloomberg emails (using an import.type=BBMail parameter), you must first extract the attachments from the attachment archive.

Decompress Attachment Archive Before Importing Bloomberg Emails

The archive file containing the attachments associated with a dump file of Bloomberg emails is a compressed tar format file. Tar files are collections of individual files collated into a single archive file.

Before starting the import operations, you must first decompress and unpack the attachments from the archive. You can then copy the individual attachment files to the import source folder specified by the parameter BBMail.DirAttachment.

More information:

Bloomberg Email Parameters (see page 69)

Event Import Parameters

The following sections describe the available parameters for Event Import.

- Import Type Parameter (see page 44)
  `Import.Type`
- Engine Parameters (see page 45)
  `Engine.BulkImportUsername`
  `Engine.BulkImportUserpasswd`
  `Engine.WorkerThreads`
  `Engine.StopOnError`
  `Engine.EventNumberInImporterHigh`
  `Engine.EventNumberInImporterLow`
  `Engine.EventRetentionPeriod`
  `Engine.LogLevel`
  `Engine.LogMaxSizeBytes`
  `Engine.LogMaxNumFiles`
  `Engine.UsePolicyEngineConnector`
  `Engine.SuppressBlobCaching`
- Email General Parameters (see page 50)
  `EMail.IgnoreAPMAuditMails`
  `EMail.InternalAddrPattern`
  `EMail.SenderAddrIncludeFilter`
  `EMail.SenderAddrExcludeFilter`
  `EMail.EventDateFromEMail`
  `EMail.StoreMessageClass`
  `Email.MessageClassFilterIncludePattern`
  `Email.MessageClassFilterExcludePattern`
  `Email.MoveToFailedFolderOnHubTimeout`
- File Handling Parameters
  `File.Pathspec`
  `File.IncludeSubdirs`
  `File.DeleteAfterImport`
  `File.ContinuousInput`
- Exchange Server Import Parameters (see page 57)
  `MSExch.ServerName`
  `MSExch.MailboxName`
  `MSExch.DeleteEMailAfterImport`
  `MSExch.ContinuousInput`
  `MSExch.IncludeMailboxFolder`
  `MSExch.FailedMailboxFolder`
  `MSExch.ArchiveConnectorName`
  `MSExch.ExpandDLs`

- NSF File Parameters (see page 62)
  NSF.DominoServerName
  NSF.DominoFileName
  NSF.ImportPassword
  NSF.OpenRetries
  NSF.RemoteDataLocationDBItem
  NSF.RemoteDataLocationType
  NSF.FolderName
  NSF.FailedMessageFolder
  NSF.DeleteAfterImport
  NSF.RetryFailedOnStartup
- PST File Parameters (see page 66)
  PSTFile.IncludePSTFolder
  PSTFile.AllowPSTPasswordUI
  PSTFile.PSTPassword
- EML File Parameters (see page 67)
  EML.RemoteDataLocationType
  EML.RemoteDataLocationMimeHeaderTag
  EML.RequiresDataLocationMimeHeaderTag
- Bloomberg Email Parameters (see page 69)
  BBMail.DirAttachment
  BBMail.DeleteAttachments
  BBMail.RecurseDirAttachment
  BBMail.RejectOnAttachmentFailure
- File Import Parameters
  ImpFile.PolicyParticipant
  ImpFile.AssociatedParticipant
  ImpFile.EventDateFromFile
  ImpFile.SourceIsNBA
- Remote CMS Import Parameters: Job Setup (see page 74)
  SQL.JobStartTime
  SQL.JobEndTime
  SQL.FailedEVFDirectory
  SQL.RunViaScheduledTask
  SQL.RecordSetSize
  SQL.EventDateFromSource
- Remote CMS Import Parameters: Primary CMS (see page 77)
  SQL.MasterCMS
  SQL.MasterCMSUserName
  SQL.MasterCMSUserPassword

- R (see page 78)emote CMS Import Parameters: D (see page 74)atabase Filters
  (see page 78)SQL.AttributeFilter
  SQL.EventType
  SQL.EventFilter
  SQL.UserFilterMode
  SQL.UserFilter
  SQL.TriggerFilter

More information:

Import Type Parameter (see page 44)
Engine Parameters (see page 45)
Email General Parameters (see page 50)
File Handling Parameters (see page 54)
Exchange Server Import Parameters (see page 57)
NSF File Parameters (see page 62)
PST File Parameters (see page 66)
EML File Parameters (see page 67)
Bloomberg Email Parameters (see page 69)
File Import Parameters (see page 71)
Remote CMS Import Parameters (see page 73)

Import Type Parameter

This single, mandatory parameter specifies the type of import operation. That is, it determines whether to import from an archive file or directly from mailboxes on a Microsoft Exchange server. This parameter is required in all import operations.

Import.Type

Important! This parameter is mandatory. Each import job must include one instance only of this parameter, either in the command line or in a configuration file.

This parameter specifies which type of message or file to import. The syntax is:

```
Import.Type=<file type>
```

Where <file type> can be:

- **BBMAIL** to import XML dump files containing emails sent using Bloomberg terminals.
- **EML** to import .EML Microsoft Internet Mail files.
- **EVF** to import .EVF CA DataMinder event files.
- **EXCH** to import from a Microsoft Exchange mailbox.
- **FILE** to import any type of files (typically text-based documents).
- **MSG** to import .MSG message files.
- **NSF** to import .NSF Lotus Notes archive files.
- **PST** to import .PST Outlook archive files.
- **SQL** to import CA DataMinder events (emails and files) from a remote CMS.

Note: If you include import.type in a command line, it overrides any instance of import.type in the configuration file. For example, this command specifies an EVF import operation, regardless of how Import.Type is assigned in the params.ini configuration file:

```
wgnimp -f params.ini -p import.type=EVF Import.Type
```

Engine Parameters

These general parameters can be used to configure any type of import operation. They cover such areas as the administrator account used to log in to the CMS, how individual events are associated with CA DataMinder user accounts (including the handling of imported events that do not match any existing CA DataMinder user), logging options and event queue handling.

Engine.UsePolicyEngineConnector

(For Import Policy jobs only) Defaults to No. This parameter specifies how to implement the Import Policy feature. The syntax is:

```
Engine.UsePolicyEngineConnector=Yes, No or Hub
```

Import Policy provides a mechanism to connect Event Import to policy engines in order to apply policy triggers to events as they are imported. You can set this parameter to any of the following:

Yes

Event Import passes events directly to the local policy engine. The policy engine analyzes these events, applying policy triggers as necessary, and then replicates the events up to the CMS. If this parameter is set to Yes, you also need to configure the Policy Engine settings in the machine policy.

No

Event Import stores events in the local database without applying policy.

Hub

Event Import passes imported events to multiple (local or remote) policy engines via a policy engine hub (the policy engine connector). When a control trigger activates, a *control event* is generated and saved on the CMS (for example, a warning or blocking). If using this setting, you also need to configure the policy engine parameters in the import.ini template file on the machine hosting the policy engine hub.

Note: If Import Policy is in hub mode, you must edit the corresponding registry values to determine which user policy to apply to imported emails.

Using the iConsole, or Data Management console, these events on the CMS can then be searched for and reviewed in the normal way. But crucially, because the email has already been sent, the *control action* can never be invoked, so warning dialogs are not actually shown, emails are not actually blocked, and so on.

Engine.BulkImportUsername

This specifies the name of an existing CA DataMinder user that is used to create new CA DataMinder users as necessary. The syntax is:

```
Engine.BulkImportUsername=<CA DataMinder user name>
```

This user must have the 'Events: Allow event import' and 'Events: Allow bulk session management' administrative privileges.

If Engine.BulkImportUsername is not specified in the command line or configuration file, Event Import prompts you for a user.

The account credentials specified by this parameter are the same as those set using the -setcredentials command line parameter.

Note: This parameter cannot be used with Import Policy.

Engine.BulkImportUserpasswd

This specifies the password for the CA DataMinder user specified by the Engine.BulkImportUsername parameter. The syntax is:

```
Engine.BulkImportUserpasswd=<CA DataMinder user password>
```

If Engine.BulkImportUserpasswd is not specified in the command line or configuration file, Event Import prompts you for a password.

Note: This parameter cannot be used with Import Policy.

Engine.WorkerThreads

Defaults to 10. This specifies the number of concurrent 'worker' threads used by Event Import to import events. The syntax is:

```
Engine.WorkerThreads=<number>
```

For Import Policy jobs, it is possible to use fewer worker threads. You only need to increase the number of worker threads if the policy engine hub is not receiving enough events to process.

Engine.StopOnError

Defaults to No. This specifies whether non-critical errors cause Event Import to stop importing events the first time an error occurs, or whether it logs the error and continues importing. The syntax is:

```
Engine.StopOnError=Yes or No
```

Engine.EventNumberInImporterHigh

Note: Formerly known as Engine.EventNumberInQueueHigh.

Defaults to 250. This parameter specifies the maximum number of events that can be open in the importer at any one time. The syntax is:

```
Engine.EventNumberInImporterHigh=<number of open events>
```

If importing is slow due to excessive memory swapping, you can reduce the number of events held in the queue. When the number of events in the queue rises to the number specified by this parameter, further event imports are suspended until the number of events in the importer has fallen below the threshold defined by Engine.EventNumberInImporterLow (see below). This allows more time to process and store the pending events.

Importing from Exchange 2003

Important! If importing emails from Exchange 2003, Engine.EventNumberInImporterHigh must **not** be set above 200! This is because of a known issue with MAPI clients opening more than the default number of server objects. For details, see MS Knowledge Base article Q830829.

Engine.EventNumberInImporterLow

Note: Formerly known as Engine.EventNumberInQueueHigh.

Defaults to 200. This parameter specifies when import operations resume. The syntax is:

```
Engine.EventNumberInImporterLow=<numberof open events>
```

If event imports are suspended because Engine.EventNumberInImporterHigh has been reached, they only resume when the queue size falls below the Engine.EventNumberInImporterLow threshold.

Engine.EventRetentionPeriod

This specifies the minimum retention period for imported events. The syntax is:

```
Engine.EventRetentionPeriod=<number of days>
```

That is, it defines how many days imported events are retained before they become earmarked for purging from the local database (typically the CMS). To specify that imported events never become eligible for purging, set this parameter to zero.

If this parameter is not set, the retention period for these imported events is determined by the Minimum Retention (Days) setting in the CMS policy.

Out of date emails are ignored

If the EMail.EventDateFromEMail parameter is set to Yes, then the capture date assigned to an imported email is derived from its delivery date or the date it was sent, *not the date it was imported*. However, the retention period is always calculated from an event's capture date. This means that for some emails, the retention may have already expired before they can be imported. These 'out of date' emails are ignored by Event Import and excluded from the import job.

For example, an import job runs on 1 June, with an event retention period of 90 days. However, the import job includes an email sent on 1 January. Because 90 days has already elapsed since 1 January, the email is ignored by Event Import.

Note: This parameter cannot be used with Import Policy.

Engine.LogLevel

Defaults to 2. This determines the level of logging for the import process. The syntax is:

```
Engine.LogLevel=<number>
```

For example, you can configure Event Import to only log errors or important system messages. The supported logging levels are:

1 Errors only

2 Errors and warnings

3 Errors and warnings, plus informational and status messages.

Log entries are written to the evtimport_<instance name>_<date>.log file, where <instance> is the name set by the Event Import service wgnimpsv.exe and <date> is the date and time when the log file was created; find this file in the system\data\logs folder.

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

Engine.LogMaxSizeBytes

Defaults to Unlimited. This specifies the maximum size for each log file. The syntax is:

```
Engine.LogMaxSizeBytes=<number of bytes>
```

When the current log file reaches its maximum size, the import process creates a new log file. Log entries are written to the evtimport_<instance name>_<date>.log file—for details see Engine.LogLevel.

Engine.LogMaxNumFiles

Defaults to Unlimited. This specifies the maximum number of log files. The syntax is:

```
Engine.LogMaxNumFiles=<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.

Engine.SuppressBlobCaching

Defaults to No. When importing emails from a third party archive, this parameter specifies whether to create temporary blob (Binary Large Object) files for each imported email. Event Import ignores this parameter when importing data from other sources, such as Exchange mailboxes or PST files. The syntax is:

```
Engine.SuppressBlobCaching=Yes or No
```

Set this parameter to Yes to prevent Event Import creating blob files. You may want to do this when importing emails from a third party archive in order to significantly boost import performance. Note that these temporary blob files are eventually deleted by event purges on the CMS.

By default, when email are imported, CA DataMinder stores the event locally, with each event comprising metadata, written to the CA DataMinder database and including general event details such an email's delivery date, and a blob file containing the email content, saved on physical media and subsequently replicated to the CMS. When importing emails from an archive, the metadata for each event automatically includes a reference to the corresponding archive entry, so there is no need to save the email content in a blob file.

More information:

Logon Requirements for the CMS - Event Import Requirements (see page 27)
Continuous Import Operations (see page 31)
Wgnimpsv.exe and Continuous Importing (see page 28)
Configure the Event Import Job (see page 92)

Email General Parameters

These parameters can be used to configure any type of email import operation. They cover such areas as how Event Import handles sender and recipient addresses, how it identifies internal emails, whether a single imported email generates separate events for each internal sender and recipient, and how it determines the capture date for each imported event.

EMail.IgnoreAPMAuditMails

Defaults to No. This parameter specifies whether Event Import and Import Policy ignore iConsole audit emails. The syntax is:

```
EMail.IgnoreAPMAuditMails=Yes or No
```

If set to:

Yes, audit emails generated by the iConsole are ignored by Import Policy and Event Import. This ensures that policy is not applied to audit emails.

No, iConsole audit emails are imported in the same way as all other e-mails.

EMail.InternalAddrPattern

This parameter specifies an address pattern that identifies e-mails as 'internal'. The syntax is:

```
EMail.InternalAddrPattern=<address pattern>
```

For outgoing e-mails, Event Import checks the recipient address; for incoming e-mails, it checks the sender address. If the e-mail address matches the address pattern specified here, Event Import flags the e-mail as 'internal'. You can then explicitly search for these internal emails in the iConsole or Data Management console.

A single Event Import operation can include multiple instances of this parameter, with each instance specifying a different way of identifying internal addresses. For example, you can specify separate address patterns for the SMTP, X.400 and EX formats (EX is the format used internally by Microsoft Outlook). The examples below show how you could use three instances of this parameter to flag all e-mails sent internally within the Unipraxis organization in the UK:

EX: EMail.InternalAddrPattern=ex:/o=unipraxis/ou=uk

SMTP: EMail.InternalAddrPattern=unipraxis.co.uk

X.400: EMail.InternalAddrPattern=c=uk;p=Unipraxis

The mechanism used by Event Import to match email addresses against specified patterns is the same as that used by CA DataMinder to match email addresses against lists of addresses defined in the capture control triggers of a user policy. For details, see the Administration console online help; search the index for 'email addresses: matching addresses to policy lists'.

This parameter is similar to the 'Internal emails' setting in the user policy, which also defines the address patterns that identify an e-mail as internal. (Find this setting policy in the System Settings\Definitions folder). This policy setting is used to flag internal emails captured directly by capture or control triggers.

Note: We strongly recommend that you only use this parameter within a configuration file. This prevents potential problems caused by spaces within the address pattern string.

Note: This parameter cannot be used with Import Policy.

EMail.SenderAddrIncludeFilter

This parameter specifies an address pattern that determines whether to import an email or not. The syntax is:

```
EMail.SenderAddrIncludeFilter=<address pattern>
```

Sender addresses that match the address pattern specified here **are** imported. For example, you can use this parameter to only import emails where the sender email is internal to your organization.

As with EMail.InternalAddrPattern (see above), a single Event Import operation can include multiple instances of this parameter, with each instance specifying a different way of identifying sender addresses.

Note: We strongly recommend that you only use this parameter within a configuration file. This prevents potential problems caused by spaces within the address pattern string.

EMail.SenderAddrExcludeFilter

This parameter specifies an address pattern that determines whether to exclude an email from the import job. The syntax is:

```
EMail.SenderAddrExcludeFilter=<address pattern>
```

Sender addresses that match the address pattern specified here are **not** imported.

As with EMail.InternalAddrPattern (see above), a single Event Import operation can include multiple instances of this parameter, with each instance specifying a different way of identifying sender addresses.

Note: We strongly recommend that you only use this parameter within a configuration file. This prevents potential problems caused by spaces within the address pattern string.

EMail.EventDateFromEMail

Defaults to Yes. This parameter specifies where the capture date assigned to **imported** events is set from. The syntax is:

```
EMail.EventDateFromEMail=Yes or No
```

If this parameter is set to:

Yes, the timestamp reflects the time and date in the email. It is based on the delivery time or time sent. If the e-mail does not contain the delivery time or time sent, Event Import sets the timestamp to the time of import.

No, the timestamp reflects the time of import.

EMail.StoreMessageClass

Defaults to Yes. This parameter specifies whether CA DataMinder extracts and stores the message class of imported emails. It applies to MSG, EVF, PST, EAS, EXCH and NSF import operations. The syntax is:

```
EMail.StoreMessageClass=Yes or No
```

If this parameter is set to:

Yes

CA DataMinder stores message classes for imported email events. For example, if an imported email has an IPM.Note.Fax message class, this is stored as an attribute of the event. In the Data Management console, this allows reviewers to search for or, more importantly, exclude from searches specific categories of email such as delivery receipts or meeting requests.

For Outlook emails, only 'non-standard' message classes are stored. The 'standard' message class is defined as 'IPM.Note'; nearly all Outlook emails have this message class. To avoid burdening the CMS database with unnecessary data, the message class is **not** stored when it is simply 'IPM.Note'.

No

CA DataMinder does not store message classes for imported emails.

Note: This import parameter is similar to the Store Email Class in the user policy. This setting stores the message class for **captured** emails. Find this setting in the System Settings policy folder.

Note: This parameter cannot be used with Import Policy.

Email.MessageClassFilterIncludePattern

This parameter has no default value. Reviewers can use this parameter to only import specific categories of email. The syntax is:

```
Email.MessageClassFilterIncludePattern=<message class>
```

For example, if a reviewer only wanted to import appointment notification e-mails, the syntax would be:

```
Email.MessageClassFilterIncludePattern=IPM.Note.Appointment
```

A single import.ini file can include multiple instances of this parameter, with each instance specifying a different pattern.

Email.MessageClassFilterExcludePattern

This parameter has no default value. Reviewers can use this parameter to exclude specific categories of email from the import operation. The syntax is:

```
Email.MessageClassFilterExcludePattern=<message class>
```

For example, a reviewer may want to exclude all faxes. In this case, the syntax would be:

```
Email.MessageClassFilterExcludePattern=IPM.*.Fax
```

A single import.ini file can include multiple instances of this parameter, with each instance specifying a different pattern.

Email.MoveToFailedFolderOnHubTimeout

Defaults to Yes. This parameter specifies how CA DataMinder handles emails that could not be imported because the policy engine hub timeout expired (see the GlobalEventTimeoutSeconds hub registry value). The parameter syntax is:

```
Email.MoveToFailedFolderOnHubTimeout=Yes or No
```

If set to:

No

CA DataMinder does not move the email to the ‘failure' folder, as specified by MSExch.FailedMailboxFolder. The email remains in the user’s Exchange mailbox. If the import operation is running in continuous mode, Event Import will continue trying to import the email.

Yes

The email will be moved to the ‘failure' folder.

More information:

Example Import Configuration File (see page 36)

File Handling Parameters

These parameters can be used to configure file handling in any type of import operation. They cover such areas as the location of the source files to be imported, and whether to search subfolders for target files and delete source files after a successful import. There is also a 'continuous import' parameter that causes Event Import to repeatedly scan the target folders for files to be imported.

File.Pathspec

This parameter has no default value. It specifies a fully-qualified path to the folder on the source machine containing the source files. The syntax is:

```
File.Pathspec=<file path>
```

For example:

```
File.Pathspec=C:\msgs\all_messages.msg or \imports\*.pst.
```

If you only specify a path, Event Import infers a file specification based on Import.Type. When specifying the path, be aware of the following:

Paths containing spaces

If you specify File.Pathspec directly in a command line and the name of a file or folder in the path contains spaces, you **must** enclose the entire parameter in "double quotes". For example:

```
wgnimp -f params.ini "File.Pathspec=C:\My messages\File1.msg"
```

Conversely, if you specify File.Pathspec in a configuration file, you must **not** enclose the path in double quotes, even if it contains spaces:

```
File.Pathspec=C:\My messages\File1.msg
```

Paths to mapped drives

Event Import only recognizes paths to mapped drives (for example, "Z:\Pickup Folder") when you run individual import operations using the Event Import utility, wgnimp.exe, from a command line.

Event Import *cannot* recognize paths to mapped drives when you run continuous import operations using the Event Import service (wgnimpsv.exe (see page 29)).

Paths containing # hash characters

If you specify File.Pathspec in a configuration file, you must edit any path containing a hash character '#' to ensure that wgnimp.exe interprets the path correctly. Specifically, you must prefix each hash with a backslash '\'. For example, change this:

```
File.Pathspec=C:\Messages\File#1.msg
```

To this:

```
File.Pathspec=C:\Messages\File\#1.msg
```

File.IncludeSubdirs

Defaults to No. This parameter specifies whether to search for matching files in subfolders below the source folder specified by File.Pathspec (see above). The syntax is:

```
File.IncludeSubdirs=Yes or No
```

The file specification is determined by the Import.Type parameter.

File.DeleteAfterImport

Defaults to No. This parameter specifies whether to delete the source files after an import operation. The syntax is:

```
File.DeleteAfterImport=Yes, No or Always
```

If set to:

No

The source files are never deleted. (If used with File.ContinuousInput=Yes, then the import operation will fail to start, as you cannot use these parameters together in such a way.)

Yes

The source file is only deleted after a successful import operation.

Always

The source file is always deleted, whether the import succeeds or not, and even if File.ContinuousInput=Yes (see below).

File.ContinuousInput

Defaults to No. This parameter specifies whether Event Import repeatedly scans for and imports files specified by File.Pathspec, or whether it shuts down after the input directories and files have been processed. The syntax is:

```
File.ContinuousInput=Yes or No
```

Continuous import is necessary when Event Import is running as a service and perpetually scanning an input directory.

Multiple event import processes can process a single input path when this parameter is set to Yes. To support this concurrency, a file is moved to a subdirectory of the input path while being processed. This prevents other instances of Event Import from trying to import it.

Important! If this parameter is set to '**No**', you must add the parameter File.IncludeSubdirs=Yes to your import.ini file.

If this parameter is set to **Yes**:

- It creates a \Failed subfolder on the source machine below the 'importing machine' folder (see the note below). Any files that fail to be imported are moved into this subfolder. But if File.DeleteAfterImport=Always (see above), then any 'failed' files are deleted and the \Failed subfolder is not created.

  Note: Each machine running Event Import creates its own subfolder below the folder specified by File.Pathspec on the source machine. This subfolder has the same name as the machine running Event Import. For example, if Event Import on machine UNI-KEEGAN fails to import a file, it creates the folder <File.Pathspec>\UNI-KEEGAN\Failed on the source machine.

- File.IncludeSubdirs is invalid and will fail if set to Yes. This is because Event Import creates its own subdirectory.

- File.DeleteAfterImport must be set to either Yes or Always. The import operation will fail if this parameter is set to No.

Exchange Server Import Parameters

These parameters are specifically for importing emails from mailboxes on Microsoft Exchange servers, that is, import operations of type EXCH (see Import.Type). They cover such areas as the names of the Exchange server and mailboxes you want to import from, the MAPI folders that you want to import, and whether the source emails are deleted after a successful import. There is also a 'continuous import' parameter that causes Event Import to repeatedly scan the target mailboxes for emails to be imported.

Note: See the 'MAPI Client and CDO 1.2.1' and mailbox requirements for EXCH import operations in Exchange and Outlook import operations.

MSExch.ServerName

Specifies the name of the Exchange server with the mailboxes you want to import. The syntax is:

```
MSExch.ServerName
```

MSExch.MailboxName

Specifies the name of the Exchange mailbox(es) you want to import. The syntax is:

```
MSExch.MailboxName=<mailbox>
```

You must supply the full Exchange mailbox name unless the last CN= component is known to be unique. If this is so, you need only specify this last component value to identify the mailbox. For example, for an Exchange address of:

```
/O=UNIPRAXIS/OU=FIRST ADMINISTRATIVE GROUP/CN=RECIPIENTS/CN=SRIMMEL
```

The command MSExch.MailboxName=SRIMMEL is sufficient for the Exchange server to resolve the address and identify the mailbox.

A single 'Event Import' import operation can include multiple instances of this parameter, each specifying a separate mailbox. All the mailboxes must reside on the same Exchange server. For example, to import the mailboxes for two users you add two lines to the import configuration file:

```
MSExch.MailboxName=SRIMMEL
MSExch.MailboxName=FSCHAEFFER
```

MSExch.DeleteEMailAfterImport

Defaults to No. Specifies whether to delete the source emails (in the specified mailbox folders) after an import operation. The syntax is:

```
MSExch.DeleteEMailAfterImport=Yes, No or Always
```

If set to:

No

The source emails are never deleted. But be aware that if MSExch.FailedMailboxFolder is specified, then emails that fail to be imported will still be moved to the 'failure' folder.

Yes

The source emails are only deleted after a successful import operation.

Always

The source emails are always deleted, whether the import succeeds or not.

MSExch.ContinuousInput

Defaults to No. Specifies whether Event Import repeatedly scans the mailboxes specified by MSExch.MailboxName, and whether it shuts down after the mailboxes have been processed. The syntax is:

```
MSExch.ContinuousInput=No or Yes,<timeout in seconds>
```

If MSExch.ContinuousInput=Yes,<timeout> then Event Import waits after importing all emails from the specified mailboxes until the specified timeout expires. It then restarts the import operation. For example:

```
MSExch.ContinuousInput=Yes,20
```

Restarts import operations after 20 seconds. If no <timeout> is specified, Event Import waits 10 seconds before restarting.

Note: When importing from a mailbox (for example, for Import Policy), you need to set MSExch.DeleteEMailAfterImport=Yes, unless you are importing for diagnostic or testing purposes.

MSExch.IncludeMailboxFolder

Specifies which MAPI folders within the mailbox to import. The syntax is:

```
MSExch.IncludeMailboxFolder=<folder_name>,Yes or No
```

If this parameter is not included, the entire contents of the mailbox are imported.

When identifying a folder, use backslashes as separators, followed by Yes or No to indicate whether subfolders are also imported. You must include a comma before the Yes or No option. If you do not specify Yes or No, you do not need a comma and MSExch.IncludeMailboxFolder defaults to Yes. The following examples illustrate the parameter usage:

```
MSExch.IncludeMailboxFolder="\Sent items",yes
```

Yes and No options are caseinsensitive.

```
MSExch.IncludeMailboxFolder=\inbox\UNIPRAXIS
```

Subfolders are imported automatically..

```
MSExch.IncludeMailboxFolder=\inbox\UNIPRAXIS,no
```

Subfolders are not imported.

A single 'Event Import' import operation can include multiple instances of this parameter, with each instance specifying a particular folder. For example, to import two mailbox folders you add two lines to the import configuration file:

```
MSExch.IncludeMailboxFolder=\inbox\Unipraxis
MSExch.IncludeMailboxFolder="\inbox\Personal mail"
```

Note: If the folder does not exist within the source mailbox, an error is written to the Event Import log file. Find this in the \system\data\log subfolder in the CA DataMinder installation folder.

MSExch.FailedMailboxFolder

Specifies the name and path of a MAPI folder used to store emails that could not be imported. The syntax is:

```
MSExch.FailedMailboxFolder=<folder name>,Yes or No
```

This 'failure' folder is created as a subfolder below the user's Exchange mailbox. When identifying a folder, use backslashes as separators, followed by Yes or No. If followed by:

Yes

The 'failure' folder is created automatically if it does not already exist.

No

The import job stops and an error is written to the Event Import log file.

You must include a comma before the Yes or No option. If you do not specify Yes or No, you do not need a comma and MSExch.FailedMailboxFolder defaults to Yes. The following examples illustrate the parameter usage:

```
MSExch.FailedMailboxFolder=\Failed,no
```

Yes and No options are case-insensitive.

```
MSExch.FailedMailboxFolder=\inbox\Failed
```

Subfolders are created automatically..

```
MSExch.FailedMailboxFolder="\Failed to import"
```

Enclose folder paths in double quotes if they include spaces.

Note: If MSExch.DeleteEMailAfterImport=Always, then emails that cannot not be imported are always deleted; they are not moved to a 'failure' folder.

MSExch.ArchiveConnectorName

Specifies a COM class name or GUID that contains an interface to a third-party archive system. The syntax is:

```
MSExch.ArchiveConnectorName=<Connector GUID or COM class>
```

The parameter enables emails imported from Exchange mailboxes to be physically stored as MAPI messages in a third-party archive system. CA DataMinder stores RDM references to the archived emails.

The required COM class name file or GUID depends on the archive system. For further details, please contact CA Support.

MSExch.ExpandDLs

Defaults to No. Specifies how Event Import handles emails sent to distribution lists. The syntax is:

```
MSExch.ExpandDLs=Yes or No,[optional expansion limit]
```

Specifically, import jobs can be optionally amended to save distribution list members as event participants. This permits reviewers to search for any emails received by an internal user, not just emails where the user was specified as an individual addressee. If this parameter is set to:

No

Event Import does not expand distribution lists.

Instead, it handles the distribution list as though it were a single recipient, saving the distribution list itself as an 'event participant' and also saving the distribution list's email address.

Unfortunately, this means you cannot search for these imported emails in the iConsole or Data Management console by specifying an individual recipient. Of course, because the sender is generally identifiable, you can still retrieve this imported email by searching for outgoing emails associated with the sender.

Yes, Not followed by expansion limit

Event Import always expands distribution lists and creates separate participants for individual list members. By default, there is no limit on the number of individual recipients that Event Import can extract from a distribution list. For example, if a distribution list contains several nested distribution lists, Event Import extracts all members of the nested lists.

Yes, followed by expansion limit

Delays can occur while the members are extracted and saved as event participants If the email was sent to a very large or heavily nested distribution list.

To alleviate this problem, you can specify an expansion limit. This causes Event Import to only expand distribution lists up to this specified limit. For example, to specify a maximum limit of 100 list members, set this parameter to:

```
MSExch.ExpandDLs=Yes,100
```

In this example, Event Import only creates separate participants for the first 100 list members.

Important! If a distribution list contains more than 100 members, any subsequent list members are *not* saved as participants!

MSExch.IgnoreNoPublicFolders

Defaults to No. Specifies whether MAPI ignores the lack of a Public Folder store on the Exchange server. The syntax is:

```
MSExch.IgnoreNoPublicFolders=Yes or No
```

No

Specifies that MAPI fails to connect to the Exchange server if there is no Public Folder store. Choose No when attaching to an Exchange 2003 server.

Yes

Specifies that MAPI ignores the lack of a Public Folder store on the Exchange server. Choose this when attaching to an Exchange 2007 (or later) server where the Public Folder store is not deployed. Do not set this option to Yes when attaching to an Exchange 2003 server.

NSF File Parameters

These parameters are specifically for importing e-mails from Lotus Notes NSF files. They identify the source Domino server and target NSF files and folders. You can also specify whether source emails are deleted after import operations. NSF import operations can also use Email general parameters) and the File.ContinuousInput parameter.

Note: Before using wgnimp.exe from a command to import .NSF files, you must adjust the logging level for the local Notes client. This suppresses unnecessary progress messages.

NSF.DominoServerName

This compulsory parameter specifies the name of the Domino Server hosting the NSF file (**not** the name of the machine hosting the Domino Server). The syntax is:

```
NSF.DominoServerName=<name of NSF host server>
```

Server names are not case-sensitive. A single Event Import operation can only include one instance of this parameter. For example, if the Domino Server name is domino-sales/unipraxis, hosted on the machine UNI-HARDY-W2K3, set this parameter to:

```
NSF.DominoServerName=domino-sales/unipraxis
```

To import a local NSF file, that is, to open import databases from the local data directory (as specified by the Directory variable in notes.ini), set this parameter to:

NSF.DominoFileName

This compulsory parameter has no default value. It specifies a path to the NSF file relative to the Domino server. The syntax is:

```
NSF.DominoFileName=<file path>
```

A single Event Import operation can include multiple instances of this parameter, with each instance specifying a separate NSF file:

```
NSF.DominoFileName=mail\sales\frankschaeffer.nsf
NSF.DominoFileName=mail\sales\spencerrimmel.nsf
```

You can also use ? and * wildcards.

NSF.ImportPassword

This optional parameter specifies the password for the Notes user on the Event Import host machine. The password is needed to access the Notes databases that you are importing from. The syntax is:

```
NSF.ImportPassword=<password>
```

To avoid a security loophole, you can configure wgncred.exe to securely cache this password; the required component ID is: **NotesImport**

Note: If the password is not set here or by using wgncred.exe, the importer exits and an incorrect password error is written to the Event Import log file. Find this in the \system\data\log subfolder in the CA DataMinder installation folder.

NSF.OpenRetries

Defaults to 100. This optional parameter determines how many times Event Import attempts to open an individual note or a Notes database if the Domino server is not responding fast enough. The syntax is:

```
NSF.OpenRetries=<number of attempts>
```

Raise this parameter to a higher value if events are failing to import.

NSF.RemoteDataLocationDBItem

This parameter specifies the Notes Mail item (that is, the named property) that contains the unique Archive ID allocated by a third party archive solution. The syntax is:

```
NSF.RemoteDataLocationDBItem=<named property>
```

You can also include in this parameter a filter expression to extract the Archive ID from any other information stored in the named property specified. For example, for integration with IBM DB2 CommonStore for Domino, set this parameter to:

```
NSF.RemoteDataLocationDBItem=CSNDArchiveID, ["DOCID\#{%until(\#)%}"]
```

This example tells Import Policy that the Archive ID is stored in the CSNDArchiveID property and gives a filter expression to extract the Archive ID from other information stored there.

NSF.RemoteDataLocationType

This parameter specifies the email archive type. The syntax is:

```
NSF.RemoteDataLocationType=<archive type>
```

For integration with IBM DB2 CommonStore for Domino, set this parameter to:

```
NSF.RemoteDataLocationType=CSLD
```

NSF.FolderName

This optional parameter specifies which folders within the NSF file you want to import. The syntax is:

```
NSF.FolderName=<name of folder>,Yes or No
```

A single Event Import operation can include multiple instances of this parameter, with each instance specifying a particular folder:

```
NSF.FolderName=Internal,no
NSF.FolderName=Internal\Memos,no
```

To import standard folders such as Inbox or Sent, prefix the folder name with a $ character and capitalize the first letter. When identifying subfolders, use backslashes as separators. Add Yes or No after the folder name to indicate whether subfolders are also imported; you must include a comma before the Yes or No option. If you do not specify Yes or No, you do not need a comma and NSF.FolderName defaults to Yes. The following examples illustrate the parameter usage:

```
NSF.FolderName=Sales,yes
```

Yes and No options are case-insensitive.

```
NSF.FolderName=$Inbox
```

Subfolders are imported automatically..

```
NSF.FolderName=Sales\North America,no
```

Subfolders are not imported.

Note: The NSF.FolderName parameters apply to all NSF.DominoFileName parameters. If a specified folder does not exist within the source NSF file, an error is written to the Event Import log file. Find this in the \system\data\log subfolder in the CA DataMinder installation folder.

Note: This parameter is unable to detect ‘unparented’ emails, that is, emails that have been moved from their original folder within the NSF file.

NSF.FailedMessageFolder

This parameter specifies the name of the view used by the importer to display failed messages in the Notes Database. The syntax is:

```
NSF.FailedMessageFolder=<view name>
```

If an email cannot be imported, the importer creates this view in the Notes Database and then adds the WiganImportStatus property to the email to mark it as 'failed'. The email is *not* moved to a separate folder.

Note: If this parameter is not specified, 'failed' emails cannot be viewed in the Notes Database.

NSF.DeleteAfterImport

Defaults to No. This optional parameter specifies whether to delete the source emails (in the specified file or folders) after they have been successfully been imported. The syntax is:

```
NSF.DeleteAfterImport=Yes or No
```

If set to:

No

The source emails are never deleted.

Yes

The source emails are deleted after a successful import operation.

NSF.RetryFailedOnStartup

Defaults to Yes. This optional parameter specifies whether messages that previously failed to import are retried when the importer is restarted. The syntex is:

```
NSF.RetryFailedOnStartup=Yes or No
```

When the importer is restarted, if this parameter is set to:

No

Event Import does not try to re-import any previously failed messages.

Yes

Event Import tries to re-import all messages that previously failed to import. If using continuous import, messages are only retried on their first pass through the database.

Event Import also removes the WiganImportStatus property from 'failed' emails to enable it to retry the import process.

If this parameter is *not* included in the import configuration file, and 'failed' messages exist, Event Import assumes a default value of Yes.

PST File Parameters

These parameters are specifically for importing e-mails from PST files. They determine which MAPI folders are imported and how Event Import handles password-protected PST files. Note also the PST import operations can also use Email general parameters and File handling parameters.

PSTFile.IncludePSTFolder

This parameter specifies which MAPI folders within the .PST file to import. The syntax is:

```
PSTFile.IncludePSTFolder=<folder_name>, Yes or No
```

If this parameter is not included, the entire contents of the .PST file are imported.

A single 'Event Import' import operation can include multiple instances of this parameter, with each instance specifying a particular folder. When identifying a folder, use backslashes as separators, followed by Yes or No to indicate whether subfolders are also imported.

You must include a comma before the Yes or No option. If you do not specify Yes or No, you do not need a comma and PSTFile.IncludePSTFolder defaults to Yes. The following examples illustrate the parameter usage:

```
PSTFile.IncludePSTFolder==Sales,yes
```

Yes and No options are case-insensitive.

```
PSTFile.IncludePSTFolder==$Inbox
```

Subfolders are imported automatically..

```
PSTFile.IncludePSTFolder==Sales\North America,no
```

Subfolders are not imported.

PSTFile.AllowPSTPasswordUI

Defaults to Yes. This parameter specifies whether to prompt the user for a password if a PST file requires one. The syntax is:

```
PSTFile.AllowPSTPasswordUI=Yes or No
```

If set to:

Yes

Event Import first tries the password specified by PSTFile.PSTPassword. If this fails, it then prompts the user.

Note: This option does not affect import operations for non-password protected .PST files. That is, any PST files that are not password protected are imported as normal.

No

Event Import attempts to import all specified PST files. If any of these files are password protected, these files are not imported and an entry is added to the event import log file.

More information:

Import Failures (see page 37)

EML File Parameters

These parameters are for importing internet e-mails from EML files. Event Import recognizes any internet e-mail file as an EML file if it conforms to RFC2822. Note that EML files, as recognized by Event Import, may have different file formats. EML import operations can also use Email general parameters and File handling parameters.

Note: RFC2822 is commonly used to mean 'Internet Message Format', based on the name of the RFC document describing it.

EML.RemoteDataLocationType

When importing EML files, the Internet email may be archived in a remote location. This parameter specifies the type of archive where the remote data is located. This information is used by the Remote Data Manager (RDM) to retrieve data stored remotely during an event search. The parameter syntax is:

```
EML.RemoteDataLocationType=<archive type>
```

When specifying this parameter, its value must match one of the supported archive types. For a list of supported archive vendors and associated parameter values, please contact CA Technical Support.

Note: If this parameter is set, you must also specify EML.RemoteDataLocationMimeHeaderTag.

EML.RemoteDataLocationMimeHeaderTag

Set this parameter to the name of the header tag in the EML file that identifies the location of the data file in the remote archive specified by EML.RemoteDataLocationType. The syntax is:

```
EML.RemoteDataLocationMimeHeaderTag=<header tag>
```

This header tag was set by the application that originally stored the email in the remote archive.

If this parameter specifies a header tag that is not present in the header of the EML file, the event will be imported with no remote data type or location and a warning is written to the CA DataMinder Event Import log.

Note: If this parameter is set, you **must** also specify EML.RemoteDataLocationType.

EML.RequiresDataLocationMimeHeaderTag

Defaults to Yes. This parameter specifies whether the specified header tag must be present in the EML files. The syntax is:

```
EML.RequiresDataLocationMimeHeaderTag=Yes or No
```

If set to:

Yes

Event Import will fail (that is, will not import) any emails that do not have the specified tag. This is also the default behavior if this parameter is not set.

No

Event Import will import any emails that do not have the specified tag, but they are saved as 'local' events (the corresponding blob file is saved on the CMS) , instead of being saved as 'remote' events with the actual message data stored in a remote archive.

More information:

Import Failures (see page 37)
Email General Parameters (see page 50)

Bloomberg Email Parameters

These parameters are specifically for importing attachments associated with Bloomberg emails (that is, emails sent using Bloomberg terminals). You use these parameters in conjunction with BBMAIL import operations; see Import.Type.

These parameters specify the folder containing any e-mail attachments and also determine whether attachments are deleted after being successfully attached to their associated email. BBMAIL import operations can also use Email general parameters and file handling parameters.

Note: The location of the XML dump file containing the actual emails is specified by File.Pathspec.

Important! Bloomberg attachments are typically supplied as a compressed archive. Before starting the import operation, you must first decompress the attachment archive.

BBMail.DirAttachment

This parameter has no default value. It specifies a fully-qualified path to the root folder on the source machine containing the attachments associated with the Bloomberg emails you are importing during a BBMAIL import operation. The syntax is:

```
BBMail.DirAttachment=<Directory>
```

Paths containing spaces

If you specify BBMail.DirAttachment directly in a command line and the name of a folder in the path contains spaces, you **must** enclose the entire parameter in "double quotes". For example:

```
wgnimp -f params.ini "BBMail.DirAttachment=C:\BBG\My Attachments"
```

Conversely, if you specify BBMail.DirAttachment in a configuration file, you must **not** enclose the path in double quotes, even if it contains spaces:

```
BBMail.DirAttachment=C:\BBG\My Attachments
```

BBMail.RecurseDirAttachment

Defaults to No. This parameter specifies whether to search for attachments in subfolders below the root folder specified by BBMail.DirAttachment. The syntax is:

```
BBMail.RecurseDirAttachment=Yes or No
```

BBMail.DeleteAttachments

Defaults to No. This parameter determines whether attachment files are deleted after they have been successfully attached to imported Bloomberg emails. The syntax is:

```
BBMail.DeleteAttachments=Yes or No
```

BBMail.RejectOnAttachmentFailure

Defaults to No. If an attachment is missing or cannot be found, this parameter determines whether the entire XML dump file is flagged as an import failure. The syntax is:

```
BBMail.RejectOnAttachmentFailure=Yes or No
```

If set to Yes, the BBMAIL import operation will stop importing from a 'failed' dump file and begin processing the next dump file.

More information:

Bloomberg Message Attachments (see page 40)
Import Failures (see page 37)

File Import Parameters

These parameters are specifically for importing files into CA DataMinder. Use them to associate CA DataMinder users with imported files and to determine how file capture dates are set. For Import Policy operations, you can also specify which user policy is applied to the imported files.

Note: Do not confuse these ImpFile.* parameters with the File.* file handling parameters, which cover such areas as the location of the source files to be imported, and whether to search subfolders for target files.

ImpFile.PolicyParticipant

No default value. This parameter identifies which user policy is applies to imported files. It is mandatory when importing files as part of an Import Policy operation. The syntax is:

```
ImpFile.PolicyParticipant=<email address>, Yes or No
```

This parameter must be set to an email address followed by Yes or No. You must include a comma before the Yes or No option. This address **must** match an address associated with a CA DataMinder user (as listed in the User Properties dialog in the Administration console).

This parameter is only used when importing files as part of an Import Policy operation. For example, you may want to apply policy to imported files in order to categorize or apply smart tags to important business documents. When the policy engine processes an imported file, it maps the specified email address to a CA DataMinder user account and applies that user's policy.

The Yes or No option determines whether this 'policy user account' is added to the list of event participants (in this case, the users associated with imported file). If set to:

Yes

The specified account **is** added to the list of event participants. Choose this option if you want to apply a specific user's policy to the imported file **and** associate that same user with the resulting file event.

No

The specified account is **not** added to the list of event participants. You typically may choose this option if the 'policy user account' is not a real person, but simply an account that you use to apply a specific set of policy triggers. For example, you may have a Compliance user account with a customized user policy designed to enforce a specific set of regulations.

If you do not specify Yes or No, you do not need a comma and ImpFile.PolicyParticipant defaults to Yes.

ImpFile.AssociatedParticipant

No default value. Use this parameter to identify any CA DataMinder user associated with, or linked to, imported files. The syntax is:

```
ImpFile.AssociatedParticipant=<email address>
```

Typically, this user will be the author of the file. For example, if importing files from a specific user's workstation or from their share on a file server, you can use this parameter to associate those files with that user.

The parameter must be set to an email address that matches an address associated with a CA DataMinder user account (as listed in the User Properties dialog in the Administration console). A single Event Import operation can only include a single instance of this parameter. For example, to associate all imported files with Spencer Rimmel, add this line to the import configuration file:

```
ImpFile.AssociatedParticipant=srimmel@unipraxis.com
```

If no associated participant is specified

All imported files are automatically associated with the machine hosting the source folder (only if the source folder is hosted locally—see the note below). Specifically, an address matching the machine's domain name in Active Directory is associated with each imported file event and stored in the CMS database. This machine 'address' takes the form /cn=<computer name>/cn=computers. For example:

/cn=UX-MILAN-W2K3/cn=computers

This means that even if this parameter, ImpFile.AssociatedParticipant, is not used, each imported file is still associated with a 'host machine' address. In this situation (based on the example above), to ensure that files imported from host machine UX-MILAN-W2K3 can be retrieved during an iConsole event search, you would need to add the above machine address to the list of addresses specified for an appropriate CA DataMinder user account. You add new addresses in the User Properties dialog in the Administration console.

Note: If the import source folder is on a network mapped drive or a UNC path, a host machine address is not created and the resulting file event is not associated with a machine address.

Note: For further details about mapping file events to CA DataMinder users, see the 'Event Participants' technical note, available from CA Support.

ImpFile.EventDateFromFile

Defaults to Yes. This parameter specifies how the capture date assigned to imported files is determined. The syntax is:

```
ImpFile.EventDateFromFile=Yes or No
```

If set to:

Yes

The timestamp reflects the time and date when the file was last modified.

No

The timestamp reflects the time of import.

ImpFile.SourceIsNBA

Defaults to No. Use this parameter to explicitly flag imported file events as being captured by the Network Boundary Agent (NBA). This enables you to search directly for NBA events in the iConsole or Data Management console. The syntax is:

```
ImpFile.SourceIsNBA=Yes or No
```

If this parameter is set to:

Yes

All imported files in the current job are flagged as NBA file events.

When you search for file events in the iConsole or Data Management console, NBA file events are identified in the Event Type column by  icons and described as ' A file moving over the network'.

No

Imported files are not differentiated by import or capture source. When you search for file events, all file events (including those captured by the File Scanning Agent or imported from Windows machines) are represented in the Event Type column by icons and described simply as 'File'.

Remote CMS Import Parameters

Remote CMS import (RCI) jobs are used to import e-mail and file events from one CMS onto another. For example, you can use RCI jobs to import warning and response e-mails from a primary CMS into a on a secondary CMS.

RCI parameters can be grouped into:

- Job setup parameters
- Primary CMS parameters
- Database filter parameters

RCI: Job Setup Parameters

These parameters specify the earliest and latest events to be included in the import operation and the location for any EVF files generated when an event cannot be imported for any reason. Other parameters specify the maximum number of events in each batch returned to Event Import for processing and whether or not the import job runs in continuous mode.

SQL.JobStartTime

Important! This parameter is mandatory for remote CMS import operations!

Defaults to zero (see below). This mandatory parameter defines a 'job start date'. The syntax is:

```
SQL.JobStartTime=<year:month:day:hour:min:sec>
```

The job start date is the earliest capture date for imported events. That is, the import job only includes events captured after this start date; events captured before this date are not imported. This example imports events captured after midnight on 11 July 2007:

```
SQL.JobStartTime=2007:07:11:00:00:00
```

If set to zero

This parameter can also be set to zero. This means the import job has no start date; all events are imported, providing their capture date falls before the 'job end date' (if specified).

```
SQL.JobStartTime=0
```

SQL.JobEndTime

Defaults to zero (see below). This parameter optionally defines a 'job end date'. The syntax is:

```
SQL.JobEndTime=<year:month:day:hour:min:sec>
```

The job end date is the latest capture date for imported events. That is, the import job only includes events captured before this end date; events captured after this date are not imported. This example imports events captured before midnight on 12 July 2007:

```
SQL.JobEndTime=2007:07:12:23:59:59
```

Note: If an import job runs before the specified 'job end date', then the job end date is automatically set to the current time.

If set to zero

This parameter can also be set to zero. This means the import job has no end date; all events are imported, regardless of capture date:

```
SQL.JobEndTime=0
```

SQL.FailedEVFDirectory

Important! This parameter is mandatory for remote CMS import operations!

This parameter specifies the folder used to store events that could not be imported. The syntax is:

```
SQL.FailedEVFDirectory=<Path>
```

These import failures can occur, for example, if a policy engine fails to process an imported event. Events that fail to be imported are written to the specified folder as EVF files (see below). For example:

```
SQL.FailedEVFDirectory="C:\Import Failures"
```

You must enclose the folder path in double quotes if it includes spaces.

Import failures saved as EVL files if unable to generate EVF files

If Event Import fails to generate an EVF file (for example, because the connection to a policy engine is lost), it will save an EVL file to this folder. EVLs are 'event link' files that point to the associated email events on the primary CMS.

Reprocessing import failures

To retry importing these failed events, you must run a EVF import job. That is, the Import.Type parameter is set to EVF; see .

SQL.RunViaScheduledTask

This parameter enables scheduled import jobs to resume from the point at which it was stopped. The syntax is:

```
SQL.RunViaScheduledTask=Yes or No
```

You can schedule remote CMS import jobs (based on wgnimp.exe) by using Windows Scheduled Tasks. This enables you to schedule import jobs to run at fixed times for a fixed duration.

If a scheduling time slot expires before an import jobs has completed, this parameter, SQL.RunViaScheduledTask, enables the import job to resume (during the next scheduled time slot) from the point at which it was stopped. If this parameter is set to:

Yes

An interrupted remote CMS import job will resume from the point that it was stopped. That is, it ignores events returned by the database query that have already been imported. If you run scheduled import jobs, you typically also set SQL.JobEndTime to zero.

Important! Be aware that if you change any of the following parameters while an import job is running, the job will restart from the beginning (see the 'No' description below for details):

```
SQL.RunViaScheduledTask
SQL.JobStartTime
SQL.JobEndTime
SQL.EventType
```

No

An interrupted import job will restart from the beginning. That is, it will attempt to import all events returned by the database query, including events already imported. This mode is appropriate for batch import jobs.

SQL.RecordSetSize

Defaults to 4000. This parameter specifies the maximum number of events in each import batch. The syntax is:

```
SQL.RecordSetSize=<Number>
```

You do not normally need to change this number. When a database query searches a remote CMS database, it returns matching events to Event Import in batches. Each batch is based on event timestamps. Grouping the query results in this way ensures optimized import performance and avoids potential problems that can occur when very large results sets are transferred across a network.

If your CMS database is very large, and the CMS host server is sufficiently powerful, you may want to increase the batch size to enable faster importing.

SQL.EventDateFromSource

Defaults to Yes. This parameter specifies the date for the event timestamp. The syntax is:

```
SQL.EventDateFromSource=Yes or No
```

If set to:

Yes

The timestamp is set to the original timestamp from the CMS database.

No

The timestamp is set to the date when the event was actually imported.

Note: This parameter is primarily used for diagnostic or testing purposes.

More information:

Remote CMS Import Failures (see page 39)
Scheduling Remote CMS Import Jobs (see page 32)

Primary CMS Parameters

These parameters identify the source CMS database and the CA logon account (user name and password) used to access this database.

Note: Because these parameters contain potentially sensitive account details, you may prefer to specify these parameters in a command line instead of saving them in an import configuration file.

SQL.MasterCMS

Important! This parameter is mandatory for remote CMS import operations!

This parameter specifies the name of the primary CMS that you want to import events from. The syntax is:

```
SQL.MasterCMS=<server name>
```

SQL.MasterCMSUserName

This parameter specifies the user name for a valid CA DataMinder administrator account on the remote CMS. The import job uses this account to access the primary CMS database. The syntax is:

```
SQL.MasterCMSUserName=<user name>
```

Administrative privileges

This CA DataMinder user must have the 'Events: View captured data' administrative privilege. We also strongly recommend that this user has the Admin: Disable Security Group Filtering administrative privilege. Without this privilege, import jobs will run significantly slower. You grant these privileges in the Administration console. Search the Administration console online help for details.

SQL.MasterCMSUserPassword

Specifies the password for the user account specified by SQL.MasterCMSUserName when connecting to the primary CMS. The syntax is:

```
SQL.MasterCMSUserPassword=<password>
```

Database Filter Parameters

These import parameters are used to query the database on the source CMS to retrieve matching events.

- **SQL snippets:** To enable Event Import to generate the necessary database queries, several of these parameters take 'SQL snippets' as their value.

  A SQL snippet is a segment of SQL code that can be incorporated into a full SQL query. Example snippets are provided below.

- **SQL syntax and single quotes:** For Oracle CMSs, certain values in SQL snippets must be enclosed in single quotes. Specifically, if the underlying database column has a VARCHAR datatype, you must use single quotes. For parameter usage examples, see SQL.TriggerFilter and SQL.AttributeFilter.

For full details of database tables, indexes and datatypes, see the *CA DataMinder Database Schema*, available from CA Technical Support.

SQL.AttributeFilter

The parameter has no default value. You can use this parameter to filter import operations by event attributes. The syntax is:

```
SQL.AttributeFilter=<SQL snippet>
```

For example, smart tags are stored in the CMS database as event attributes. To filter import operations by event attributes, you must set this parameter to an appropriate SQL snippet that queries the AttrType and AttrValue columns in the Wgn3EA database table.

Important! You **must** use ea. as the column prefix. For example, you cannot use extendedattribute. as the column prefix because this is not supported!

SQL.EventType

Important! This parameter is mandatory for remote CMS import operations!

The parameter has no default value. It specifies whether to import emails or files. The syntax is:

```
SQL.EventType=File or EMail
```

A Remote CMS import job can import either emails or file events; a single job cannot import both. If this parameter is set to:

EMail

The import job only imports email events (and attachments, if present).

File

The import job only imports file events.

Be aware that if you configure the Remote CMS Import job to import files, you **must** also include the ImpFile.PolicyParticipant parameter. This parameter, mandatory for file import jobs, identifies which user policy to apply to imported files.

SQL.EventFilter

This parameter provides additional flexibility to filter import operations by querying the Wgn3Event database table. The syntax is:

```
SQL.EventFilter=<SQL snippet>
```

The Wgn3Event table contains all events for all users. For example, to import IM conversations embedded in emails (see the note below), set this parameter to an appropriate SQL snippet that references such events in the EventMinorType column:

```
SQL.EventFilter=e.EventMinorType=32
```

Where 32 identifies emails that contain embedded IM conversations. You *must* use e. as the column prefix; do not use event. as the column prefix because this is not supported!

Important! This parameter must be consistent with the SQL.EventType parameter (see above). If that parameter specifies an email import operation, then SQL.EventFilter must also specify a type of email event as the import filter.

Note: CA utilities such as Cnv2email.exe or BB2email.exe can convert various types of message data, including IM dump files, into EML files (that is, Internet emails) that can be subsequently processed by policy engines and imported into a CMS.

SQL.UserFilterMode

This parameter has no default value. You use this parameter in combination with SQL.UserFilter to filter import operations by **user or group**. The syntax is:

```
SQL.UserFilterMode=<Number>
```

This parameter defines the filtering mode. That is, it specifies whether the import operation will filter by names, account IDs, or email addresses, and whether to import from subgroups. It can take one of the following values:

0 No filtering

1 User ID

2 Group ID

3 Group ID, plus subgroups

4 User name

5 Group name

6 Group name, plus subgroups

7 Email address

SQL.UserFilter

This parameter is used in combination with SQL.UserFilterMode to filter import operations by **user or group**. The syntax is:

```
SQL.UserFilter=<List of names, addresses or IDs>
```

Set this parameter to a comma-separated list of user names, group names or email addresses to filter import operations:

- **User name:** If SQL.UserFilterMode=4, set this parameter to a comma-delimited list of user account names. You can also use _ and % wildcards. For example:

  ```
  SQL.UserFilter=unipraxis%,fschaeffer
  ```

- **Group name:** If SQL.UserFilterMode=5 or 6, set this parameter to a comma-delimited list of user groups. You can also use _ and % wildcards. For example:

  ```
  SQL.UserFilter=Equity Research,Sales
  ```

- **User email address:** If SQL.UserFilterMode=7, set this parameter to a comma-delimited list of email addresses. You can also use % wildcards to specify address patterns.

  This parameter queries the Wgn3Address database table. This table contains all addresses associated with CA DataMinder users. These addresses can be in any format, such as SMTP, EX or Bloomberg alias addresses. This example filters on two specific SMTP addresses:

  ```
  SQL.UserFilter=srimmel@ unipraxis.com,fschaeffer@unipraxis.com
  ```

  Alternatively, you can specify 'universal' addresses by enclosing users' names in wildcards and omitting protocol-specific elements such as @ characters. This example filters on any emails sent to or from Spencer Rimmel or Frank Schaeffer, regardless of address format:

  ```
  SQL.UserFilter=%rimmel%,%schaeffer%
  ```

- **Account IDs**

  For maximum flexibility, you can also filter import operations by unique user or group ID. Note that these account IDs can only be viewed in the CMS database; they are not displayed in the Administration console.

- **User ID** or **Group ID:** If SQL.UserFilterMode is set to 1, 2 or 3, set this parameter to a list of user or group IDs. Use the | pipe character to delimit the list. The syntax is:

  ```
  SQL.UserFilter=<IDM>,<ID>|<IDM>,<ID>|<IDM>,<ID>
  ```

  Where <IDM> and <ID> are part-keys stored in the UserIDM and UserID columns (Wgn3User table) or the GroupIDM and GroupID columns (Wgn3UserGroup table), that together uniquely identify an individual user or group. For example:

  ```
  SQL.UserFilter=1,2054|1,3397
  ```

SQL.TriggerFilter

The parameter has no default value. You use this parameter to filter import operations by policy trigger. The syntax is:

```
SQL.TriggerFilter=<SQL snippet>
```

That is, only events captured by specific triggers are imported. To filter import operations by trigger, you must set this parameter to an appropriate SQL snippet that (typically) queries the TriggerType or TriggerName columns in the Wgn3Trigger database table.

For example, to query the CMS database for events associated with a trigger named 'Custodian Responses', set this parameter to:

```
SQL.TriggerFilter=t.TriggerName='Custodian Responses'
```

Where Custodian Responses is a user-defined trigger name—see below. You *must* use t. as the column prefix; do not use trigger. as the column prefix because this is not supported!

Similarly, to query the CMS database for events associated with a specific trigger type (in this case, triggers to detect outgoing emails that match a specific document classification), set this parameter to the appropriate number:

```
SQL.TriggerFilter=t.TriggerType=34144258
```

Where 34144258 identifies 'document classifier' trigger types. For more on trigger types, see below.

Trigger names

Be aware that the TriggerName database column contains names defined by the policy administrator using the Trigger Name policy setting. It does **not** contain the non-editable trigger identifiers shown in the left-hand pane of the User Policy editor, such as 'Document Classifier 1' or 'Search Text 2'.

Trigger types

A full list of trigger types is available in the *CA DataMinder Database Schema*. However, that document lists the hexadecimal value of each trigger type, but you must use the equivalent decimal values when you specify the SQL.TriggerFilter import parameter. For example, 'document classifier' trigger types have a hexadecimal value of 0x02090002; this converts to 34144258 in decimal.

Chapter 3: Import Policy

This section introduces the Import Policy utility. Import Policy connects Event Import to policy engines in order to apply triggers to emails or files as they are imported:

For emails, Import Policy provides organizations with a full compliance review capability that is not dependent on a preventative pre-review strategy for filtering e-mail communications at source. And because Import Policy requires no integration with production e-mail systems, there is no risk of disruption to e-mail activity.

For files, Import Policy enables organizations to apply policy to any files stored on their network. For example, organizations can categorize, or apply smart tags to, important business documents or reports.

This section contains the following topics:

Direct Mode and Hub Mode

Import Policy can operate in two modes:

Direct mode

In this mode, Event Import passes events directly to a local policy engine. This is the simplest way to deploy Import Policy.

Hub mode

This mode is more complex to deploy, but allows you to use multiple policy engines to process imported emails, and to specify active and standby policy engines (see Active and standby policy engines). In this mode, Event Import passes e-mails to a policy engine connector (that is, a hub), which in turn assigns e-mails to an available policy engine.

Both modes are illustrated in the Architecture Diagrams section. See the reference below.

More information:

Architecture Diagrams (see page 86)

Which Imported Emails Are Converted into Events?

During Import Policy operations, imported emails are only saved as CA DataMinder events if they cause a policy trigger to activate. If an email does not cause a trigger to activate, *the email is discarded and is not saved as a CA DataMinder event!*

The Which Email Source? setting in the user policy allows you to disable a trigger based on the e-mail source. Two of the available options in this setting are designed explicitly for use with Import Policy jobs:

- Microsoft Exchange Server (Mailbox)
- Archive File Importers

For example, as part of an Import Policy job, you may want to generate events for all e-mails imported from an Exchange mailbox, but only generate events for certain categories of email imported from PST archive files. To achieve this, you would configure two versions of each policy trigger that you want to use: in the first trigger, Which Email Source? is configured so that the trigger can only fire when processing e-mails imported from Exchange; in the second trigger, Which E-mail Source? specifies that the trigger can only fire when processing relevant e-mails imported from an archive file.

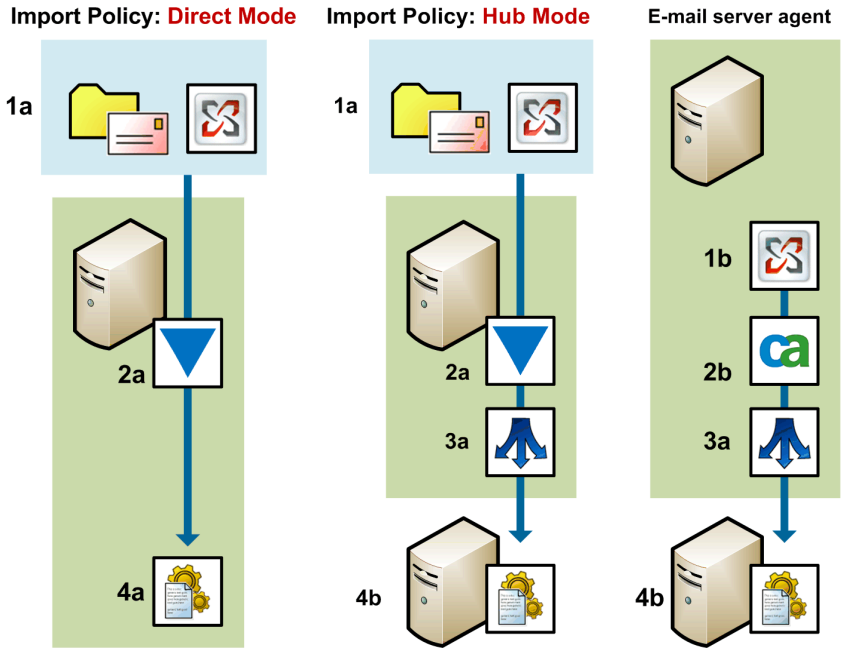
Import Policy versus Server Agents

Import Policy has parallels with the Microsoft Exchange integration provided by the Exchange server agent (see Deploy policy engines). In both cases, events from an external source are passed to a policy engine.

But there is a key difference. Under Import Policy, when a control trigger activates, a *control event* is generated and saved on the CMS (for example, a warning or blocking). Using the iConsole, or Data Management console, these events can be searched for and reviewed in the normal way. But crucially, because the email has already been sent, the *control action* can never be invoked so warning dialogs are not actually shown, e-mails are not actually blocked, and so on.

Also, because Import Policy requires no integration with production email systems, there is no risk of disruption to end-users' email activity. Import Policy also eliminates the need for CA DataMinder client agents on the desktop and policy engines on the email server.

Note: The Exchange server agent requires Microsoft Exchange Server 2003 or 2007. If your organization uses an earlier version of Microsoft Exchange, you may want to use Import Policy as a substitute mechanism for monitoring email activity.



Import Policy versus email server agents

Email ingestion

Under Import Policy, source data is extracted from Exchange journal mailboxes or email archive files (**1a**) then converted into CA DataMinder events by Event Import (**2a**). Conversely, email server agents (**2b**) simply intercept emails transiting through Exchange or Domino servers (**1b**).

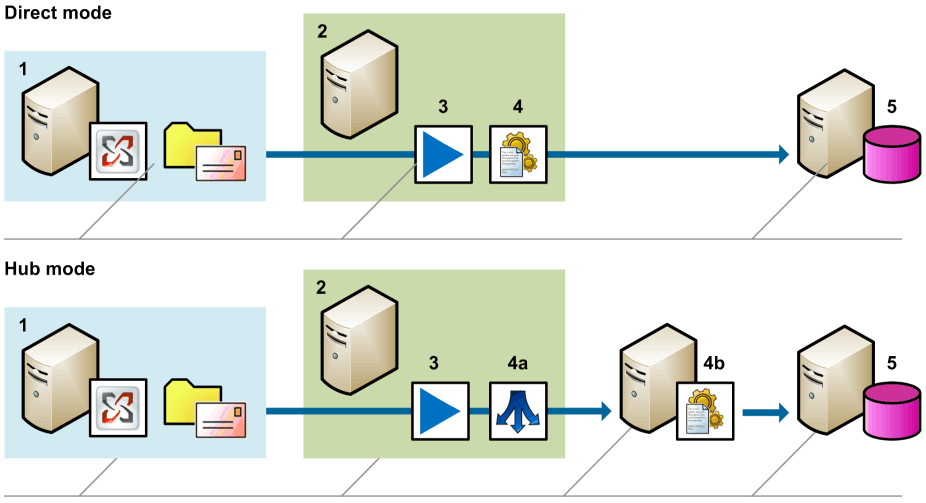
Email processing

In all cases, imported or intercepted emails are passed to a policy engine for processing. In direct mode, Import Policy passes imported emails straight to a local policy engine (**4a**). Conversely, Import Policy in hub mode passes e-mails to a local policy engine connector (**3a**). This is identical to how email server agents work, passing emails to a local policy engine hub (**3b**). In both cases, the policy engine connector and hub then allocates emails to a policy engine, typically on a remote server (**4b**). In all cases, the policy engine then applies capture and control triggers as needed.

Architecture Diagrams

The diagrams below compare the deployment architecture for Import Policy in direct mode and hub mode. Direct mode offers a much simplified deployment, while hub mode offers greater flexibility in terms of allocating emails across multiple policy engines.

The diagram below shows a hub mode deployment based on a single remote policy engine. If required, however, you can configure the policy engine connector to allocate emails across multiple policy engines.



Import Policy data flow

1. **Source data:** In both modes, emails are extracted from Exchange journal mailboxes, Notes databases or e-mail archive files.
2. **Import Policy server:** This server hosts Event Import (**3**), which converts the source data into CA DataMinder e-mail events. In direct mode, Event Import passes these events to a policy engine (**4)**. In hub mode, it passes them to a local Policy Engine Connector (**4a**). The connector then passes emails to a policy engine (**4b**), typically running on a remote server.
3. **CMS:** The policy engine analyzes the emails and applies policy triggers as necessary. After processing, email events are replicated to the CMS.

Direct Mode

This section describes how to install and deploy Import Policy in Direct mode. This is the simplest way to run the Import Policy utility, as it enables Event Import to pass events directly to the local policy engine. The policy engine analyzes these events, applying policy triggers as necessary, and then replicates the events up to the CMS.

Deploying Import Policy in direct mode is a simple two-step process. First, you must install Event Import and a policy engine on the same server, using the CA DataMinder server installation wizard—see below. Then, you must configure Event Import to pass imported emails directly to the policy engine.

Note: If a single policy engine would be too slow to process the anticipated data volumes, you can deploy Import Policy on multiple servers, with each instance importing and processing e-mails from a separate Exchange mailbox or Notes database.

Install Import Policy in Direct Mode

To install import policy in direct mode

1. Launch the installation wizard by running setup.exe. Find this in the \Server folder on your CA DataMinder distribution media.
2. In the Custom Setup screen, choose the Policy Engine and Event Import features. Note that you do not need to install the Remote PE Connector for Event Import.
3. The installation wizard now has all the information it needs. Click Install to start the file transfer.

More information:

Event Import Requirements (see page 23)

Configure the Event Import Job

To connect Event Import directly to a policy engine, you must configure parameters in the import configuration file, import.ini, and start the import job.

More information:

Event Import Parameters (see page 41)

Hub Mode

In hub mode, Import Policy connects Event Import to policy engines via a policy engine connector (or hub). Deploying import policy in hub mode is slightly more complex than Direct mode, but allows you to use multiple policy engines to process imported emails, and to specify active and standby policy engines.

Deploying Import Policy in hub mode requires the following steps

1. Before you can connect a policy engine to Event Import, you must specify a Windows domain user that allows the Policy Engine Connector and its policy engines to communicate. You must also create a corresponding CA DataMinder user account.
2. Install at least one active policy engine and, optionally, a standby policy engine
3. Install Event Import and the Policy Engine Connector.
4. Manually configure the Policy Engine Connector to enable communication between it and the policy engine(s). This involves registry changes and changes to the logon account for the hub service.
5. Configure Event Import to pass imported emails to a policy engine. This involves a parameter change.

More information;

Direct Mode (see page 87)
Specify User Accounts (see page 89)
Install Import Policy in Hub Mode (see page 90)
Configure the Remote PE Connector (see page 90)
Configure the Event Import Job (see page 92)

Specify User Accounts

In contrast to Direct mode, running Import Policy in Hub mode requires communication between the machines running the policy engines and the policy engine hub.
To implement this communication, before installing the Policy Engine Connector, you must specify a PE domain user and create a corresponding CA DataMinder user account.

PE domain user

Policy engines and the policy engine hub (in this case, the Policy Engine Connector) use the same account to communicate with each other. This account is the 'PE domain user'. This can be a new or existing domain user.

The Policy Engine Connector uses the PE domain user account to access remote policy engine machines. Policy engines use the PE domain user as their service logon account. Policy engines also use this account (or more accurately, a CA DataMinder account with the same name) to log on to the CMS.

For policy engines, you can specify the PE domain user as the service logon account when you run the installation wizard. For the Policy Engine Connector, you must manually provide the account credentials for the PE domain user *after installation*.

CA DataMinder user

After specifying the PE domain user, you must create a matching CA DataMinder user account. The policy engine uses this account to log on to the CMS when mapping email addresses to CA DataMinder users.

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 Import Policy in Hub Mode

To install Import Policy in hub mode, run the CA DataMinder server installation wizard

1. To launch the installation wizard, run setup.exe. Find this in the \Server folder on your CA DataMinder distribution media.
2. In the Custom Setup screen, choose Event Import and the Remote Policy Engine Connector. The Remote Policy Engine (PE) Connector functions as a policy engine hub and connects Event Import to one or more policy engines.
3. In the Service Accounts screen, specify the logon accounts used by the local CA DataMinder infrastructure service and the policy engine service
4. The installation wizard now has all the information it needs. Click Install to start the file transfer.

   This completes the installation process.

More information:

Configure the Remote PE Connector (see page 90)

Configure the Remote PE Connector

As stated earlier, the Remote PE Connector functions as a policy engine hub. The procedure for configuring the Remote PE Connector is the same as the standard hub configuration. That is, you must set the credentials for the PE domain user, assign a logon privilege to this account, and modify certain registry values. For details on installing and configuring policy engine hubs.

More information:

Policy Engine Hubs (see page 295)

Set the Credentials for the PE Domain User

First, you must configure the PE Connector *service* so it can log on to remote policy engine machines as the PE domain user.

To set the credentials for the PE domain user

1. On the Event Import host machine, go to the \System subfolder in the CA DataMinder installation folder.
2. From a command prompt, run:

   ```
   wgnphub -SetCredentials
   ```

   You must be logged on as an administrator to run this command.
3. The command will prompt for a user name and password. Enter the credentials of the **PE domain user**.

Note: Even if the PE Connector and policy engine are on the same machine, we still recommend that provide the connector service with the **additional** credentials of the PE domain user. This allows the PE Connector to communicate with any remote machines hosting, for example, your standby policy engine.

More information:

Specify User Accounts (see page 89)

Assign the 'Log on as a batch job' Privilege

Next, you must assign this account privilege to the PE domain user on the host machine for the PE Connector. To do this, open the Local Security Policy applet or, if the host machine is a domain controller, the Domain Controller Security Policy applet. Both applets are available in Administrative Tools.

More information:

Assign Security Privilege to the PE Domain User (see page 303)

Configure the PE Hub - Import Policy

You do this by modifying the associated registry values on the host machine. These registry values determine how it handles e-mail addresses, how much memory is allocated to its event queue, and so on. These registry values are located in the following registry key:

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

In particular, you must edit the registry values ActivePolicyEngines and (if required) StandbyPolicyEngines to include the name or IP address of the policy engine host machines.

Configure the Event Import Job

The final step in connecting a policy engine to Event Import is to configure the import parameters and start the import job.

When connecting a policy engine to Event Import, you need to configure Event Import parameters in the import configuration file, import.ini. These parameters must cover the usual areas (specifying the source data for the import job, and so on).

Required Import Parameter

In particular, Import Policy jobs must include the following parameter:

```
Engine.UsePolicyEngineConnector=Yes
```

This parameter specifies whether imported events are sent directly to a local policy engine or to a remote policy engine via the policy engine connector. Set this parameter to:

Yes to configure Import Policy to use direct mode.

Hub to configure Import Policy to run in hub mode.

Parameters not Supported for Import Policy

You **cannot** use these parameters with Import Policy:

```
EMail.InternalAddrPattern
Engine.BulkImportUsername
Engine.BulkImportUserpasswd
Engine.EventRetentionPeriod
EMail.StoreMessageClass
```

More information:

Engine Parameters (see page 45)
Event Import Parameters (see page 41)

Chapter 4: IM Import

This section introduces the IM Import utility, IMFrontEnd.exe.

This section contains the following topics:

Supported IM Formats

IMFrontEnd.exe supports these dump file formats:

Actiance (previously FaceTime)

A proprietary XML format for Actiance IM data.

CA DataMinder requires Actiance version 3 dump files. These XML dump files start with a <interactionV3> node.

Note: Despite the name change from FaceTime to Actiance, the corresponding IM Import data source parameter is still DirFaceTime.

DirIBXML

A proprietary XML format for Instant Bloomberg messages.

FactSet 1.1

A proprietary XML format for FactSet IM data.

IMlogic

A proprietary XML format for IMlogic instant messaging data. The corresponding data source parameter is DirIMLogic.

Note: You must first configure IM Manager to convert IMlogic dump files into a format supported by IMFrontEnd.exe.

More information:

Configure IMlogic Dump Files (see page 107)

IM Import Requirements

The following are IM Import requirements:

IM Import

IMFrontEnd.exe does not need to be installed on a CA DataMinder machine.

Note: For a list of supported operating systems, see the Requirements chapter in the *Platform Deployment Guide*, sections 'CMS and Gateways' and 'Endpoint Agents'.

IMlogic dump files

These must be correctly configured before starting import operations.

More information:

Configure IMlogic Dump Files (see page 107)

Deploy IM Import

This section describes how to deploy IM Import. The key deployment tasks are:

1. Install IMFrontEnd.exe. To do this, use the Integration Agents installation wizard.
2. Configure IM Import by specifying parameters in the configuration file IMFrontEnd.ini.

See the following sections for details on how IM Import assigns IM conversations to users and how it assigns IM networks.

Mapping IM Conversations to Users

Cnv2EML.exe extracts IM conversations from CNV files and embeds them within EML files (Internet emails). When these EML files are imported onto the CMS, CA DataMinder uses the following methods to associate the IM conversation with participants and to determine which user policy is applied:

Searching by user

Providing that the participants' e-mail addresses were available in the original IM dump file, the addresses for each participant in an IM conversation are stored as recipients in the EML email.

Which user policy is applied

The Cnv2EML.exe conversion parameter EML.From determines which policy is applied to the EML email. This parameter specifies an email address that the policy engine (PE) then compares against a list of internal address patterns. If the address is deemed internal and it can be mapped onto an existing CA DataMinder user account, the PE applies that user account's policy. If no matching CA DataMinder user can be found, or if the sender is deemed to be external, the policy engine applies the Unknown Internal Sender or External Sender policies.

How the IM Network Is Assigned

IMFrontEnd.exe automatically adds the IM network to each CNV file that it generates. It uses two methods, depending on the dump file format.

For IM conversations extracted from Bloomberg dump files, IMFrontEnd.exe automatically sets the IM network when processing dump files generated by these networks (that is, when using the DirIBXML parameter). This IM network is saved in the resulting CNV file.

For IM conversations extracted from dump files generated by 'gateway' IM applications such as Actiance (formerly FaceTime), the dump file contains an IM network identifier. That is, the IM network is saved as an attribute of the IM conversation. IMFrontEnd.exe automatically detects this attribute and saves the IM network in the resulting CNV file.

When the CNV files are subsequently converted to EML emails by Cnv2email.exe, the IM network is saved as an x-header in the EML email. Policy engines are configured to automatically detect and process this x-header and extract the IM network. For details, see the next section.

Embedding IM Conversations in Emails

CA DataMinder provides the Cnv2email.exe utility to convert CNV files generated by IMFrontEnd.exe into EML emails that can then be processed by policy engines and stored in email archives.

If you import your IM conversations as 'embedded IM' events (that is, IM conversations embedded in EML emails), you need to configure policy engines to correctly process these events. Specifically, policy engines need to detect and process x-headers in the EML email that contain essential IM data.

Why Store the IM Network in an x-header?

When IM conversations are imported directly into the CMS (that is, when CNV files are imported by Event Import), all associated IM data, such as participant addresses and the name of the IM network, are also imported from the CNV file and stored as attributes of the resulting IM events.

But when IM conversations are converted from CNV files to EML e-mails and saved on the CMS as 'embedded IM' events, these additional IM details cannot be saved as conventional email attributes. To overcome this problem, Cnv2email.exe stores participant addresses and the IM network as bespoke x-headers in the EML email. Policy engines automatically detect these x-headers when processing the EML email and store the relevant details with the resulting IM event.

Install IM Import

The process of extracting archived IM conversations uses IMFrontEnd.exe. You install this utility using the CA DataMinder Integration Agents installation wizard.

To install IMFrontEnd.exe

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 Bloomberg and IM Support.

   IMFrontEnd.exe is installed as part of this feature.

6. In the final wizard screen, click Install to start the file transfer.

   A sample configuration file, IMFrontEnd.ini, is installed to the \Import subfolder of the CA DataMinder installation folder. This file contains various commonly used parameters, but you may need to customize this file to suit your organization.

7. Configure IMFrontEnd.ini to suit your network environment.

Configure IM Import

You configure IM Import operations by specifying import parameters in configuration file, IMFrontEnd.ini.

To configure IM Import operations

1. Locate the IMFrontEnd.ini.

   A sample file is created automatically IMImport subfolder when you install IMFrontEnd.exe.

2. Specify the import parameters in this configuration file.

3. Run IMFrontEnd.exe.

   Note: If you change any import parameters after IMFrontEnd.exe has started to run, you must stop and restart it for your parameter changes to take effect. To stop IMFrontEnd.exe correctly, press Ctrl+C when the command window is active.

IM Import Parameters

IM Import supports the following parameters in the configuration file IMFrontEnd.ini.

DirFaceTime, DirIBXML, DirIMlogic

The parameters above have no default value. At least one must exist and all can co-exist in any order. They can specify either shortcuts, or fully-qualified paths to the folders containing the data files for each source. In each case, the syntax follows this pattern:

```
DirIBXML=<unique file path>
```

You can do one of the following:

- Specify the path to a real directory and copy shortcuts to files into the directory. For example:

  ```
  DirFaceTime=C:\IM Import\Source data\FaceTime
  ```

- Specify the full path to a file shortcut. For example:

  ```
  DirIMlogic=C:\IM Import\Source data\June_2009.lnk
  ```

  Where June_2009.lnk is a shortcut to a source data file stored elsewhere. This means that you can reconfigure the setup without editing the ini file.

Each parameter specified must contain a unique directory name. We recommend that you name these source folders so the respective client types can be easily identified. For example:

```
DirFaceTime=C:\IM Import\Source data\FaceTime
DirIBXML=C:\IM Import\Source data\DirIBXML
DirIMlogic=C:\IM Import\Source data\IMlogic
```

Note: Do not use quote marks around paths, even if they contain spaces.

Note: Despite the name change from FaceTime to Actiance, the corresponding IM Import data source parameter is still DirFaceTime.

ConsoleOutput

Defaults to Yes. This parameter determines whether to write progress messages to the command window. The syntax is:

```
ConsoleOutput=Yes or No
```

If set to Yes, progress messages **are** written to the command window; if set to No, they are **not**. This parameter can operate concurrently with the LogFile parameter (see below).

DeleteAttachments

Defaults to No. This parameter specifies whether to delete email attachments from the source folder after they have been imported. The syntax is:

```
DeleteAttachments=Yes or No
```

If set to Yes, attachments are deleted; if set to No, they are not deleted.

Note: Attachment source folders are defined by the DirAttachment parameter.

DirAttachment

This parameter has no default value. It can specify either a shortcut, or a fully-qualified path to a folder containing attachments. The syntax is:

```
DirAttachment=<file path>
```

This parameter can specify:

- A folder containing the source attachment files for imported conversations and emails. For example:

  ```
  DirAttachment=c:\im import\source data
  ```

- Shortcuts to folders containing attachment files. You can specify a separate shortcut for each source dump file. The shortcut must have the same name as the dump file to which these attachments belong. For example:

  ```
  DirAttachment=c:\im import\source data\June_2009.lnk
  ```

  Where June_2009.lnk is a shortcut to a completely different source data file stored elsewhere. This means that you can reconfigure the setup without editing the ini file.

- Subfolders containing attachment files where the name of the subfolder matches the name of the dump file to which these attachments belong.

- A folder containing the source attachment files plus any subfolders (if RecurseDirAttachment=Yes). We do not recommend this configuration, as it is the least efficient option.

Note: Do not use quote marks around pathnames, even if they contain spaces.

DirFinalDest

This parameter has no default value. It specifies the final destination folder for the .CNV files; from here, Event Import imports them into CA DataMinder. The syntax is:

```
DirFinalDest=<file path>
```

Where <file path> specifies the path and name of the final destination folder. For example:

```
DirFinalDest=C:\CA\IM Import\conversation\complete
```

Note: The .CNV files are created in a subfolder under the DirFinalDest folder.

Note: To allow automated and unattended import operations, the DirFinalDest parameter **must** match a location specified by the Event Import parameter File.Pathspec.

FaceTimeIDConversionExpression, IMlogicIDConversionExpression

These parameters convert a participant's internal ID (that is, the ID used by the source IM system) into an ID that is guaranteed to be unique. In each case, the syntax follows this pattern:

```
FaceTimeIDConversionExpression=<expression>
```

Where <expression> specifies an ID conversion expression. Typically, you use these parameters if no email address is available for a participant in an imported conversation. For example, the following expression prefixes each extracted participant ID with 'unipraxis:'

```
FaceTimeIDConversionExpression=unipraxis:["?%untilEnd%"]
```

In this case, the participant ID lyndasteel is converted to unipraxis:lyndasteel. The conversion expressions use the same syntax as the conversion expressions used by Account Import when importing LDAP attributes. They can parse, extract and (if necessary) remove or substitute any characters in the participant ID.

Note: Despite the name change from FaceTime to Actiance, the corresponding IM Import data source parameter is still FaceTimeIDConversionExpression.

IBXML.UseConversationIDAsSubject

(Applies to Instant Bloomberg DirIBXML messages only)

Defaults to No. This parameter specifies whether to use the conversationID as the conversation title. The syntax is:

```
IBXML.UseConversationIDAsSubject=<Yes or No>
```

Note: CNV2email.exe uses the conversation title as the email subject.

IBXML.UseCorporateAddresses

(Applies to Instant Bloomberg DirIBXML messages only)

Defaults to No. This parameter specifies whether to use corporate email addresses for participants instead of Bloomberg addresses. The syntax is:

```
IBXML.UseCorporateAddresses=<Yes or No>
```

If set to Yes, IM Import uses corporate addresses, if available. If set to No, IM Import uses Bloomberg addresses.

If set to Yes but a corporate address is not present, IM Import uses the Bloomberg address.

KeepIntermediatesOnFailure

Defaults to No. This parameter determines whether a failed IMFrontEnd.exe extraction keeps the intermediate files created before the extraction failed. The syntax is:

```
KeepIntermediatesOnFailure=<Yes or No>
```

This parameter can be useful when looking into why the extraction failed. These intermediate files are left in subfolders of the final destination folder (specified by the DirFinalDest parameter) and have an extension of .incomplete on the subfolder name. For example:

```
DirFinalDest=C:\CA\IM Import\conversation\July.incomplete
```

Note: Do not use this parameter if the Event Import parameters include File.IncludeSubDirs=Yes and File.ContinuousInput=No.

LogFile

This optional parameter has no default value. It appends progress messages to a specified log file. The syntax is:

```
LogFile=<file path>
```

Where <file path> specifies the path and name of the log file. For example:

```
LogFile=C:\CA\IM Import\Logs\IMFrontEnd.log
```

Note: Do not use quote marks around pathnames, even if they contain spaces.

LogFileThreshold

Defaults to 102400 bytes. This optional parameter specifies a threshold size (in bytes) for log files. When this threshold is exceeded, IM Import automatically creates a new log file. The syntax is:

```
LogFileThreshold=<number of bytes>
```

Note: The default log file size (equivalent to 100KB) also represents the minimum log file size. If this parameter specifies a threshold size smaller than the default, IM Import disregards the specified threshold and instead uses the default threshold.

MaxFiles

Defaults to 200. This parameter specifies the maximum number of .CNV output files that can be kept open simultaneously. In effect, it specifies the size of the .CNV file cache. The syntax is:

```
MaxFiles=<number of files>
```

When IM Import extracts IM conversations from dump files, it saves individual conversations to separate .CNV files. For performance purposes, this parameter prevents an unlimited number of .CNV files being open at one time. If this limit has already been reached, but IM Import needs to write details to a further .CNV file, it closes an open .CNV file and opens the required one.

Note: If IMFrontEnd.exe performance seems slow when processing large IM dump files, you may need to increase the size of the .CNV file cache, say, to MaxFiles=1000. This because the internal structure of some dump files means that IMFrontEnd.exe needs to continually access a very large number of separate .CNV files.

RecurseDirAttachment

Defaults to No. This parameter determines whether subfolders below the specified DirAttachment folder are included when IMFrontEnd.exe is searching for attachment data files. The syntax is:

```
RecurseDirAttachment=<Yes or No>
```

RejectOnAttachmentFailure

Defaults to No. This parameter specifies whether IMFrontEnd.exe ignores missing attachments and continues with the import process. The syntax is:

```
RejectOnAttachmentFailure=<Yes or No>
```

An attachment file can sometimes fail to be added to a conversation. For example, IMFrontEnd.exe may not be able to find an attachment referenced in the log file being imported. This parameter specifies whether the IMFrontEnd.exe process ignores the missing attachment and continues with the import process, or fails this log file and moves on to the next. If set to:

Yes

IMFrontEnd.exe fails the attachment's log file and moves it to the \Failed subfolder.

No

IMFrontEnd.exe ignores the failed attachment and continues to import the current log file.

RunOnce

This optional parameter has no default value. It specifies whether the IM Import process runs continuously or just once. The syntax is:

```
RunOnce=<Yes or No>
```

If set to Yes, the import process terminates after processing the import files. If set to No, the import process runs continuously.

Note: If IM Import does not find this parameter in IMFrontEnd.ini, the import process runs continuously.

ShowMessageProgress

Defaults to Yes. This parameter provides progress feedback during conversation extraction. The syntax is:

```
ShowMessageProgress=<Yes or No>
```

If set to Yes, a time-stamped progress message is written to the log file and command window, as configured, every two minutes. If set to No, progress messages are not written to the log file or command window.

ViewSize

Defaults to 250 KB (256000 bytes). This parameter specifies a maximum size for a single 'event' in an IM Import file. The syntax is:

```
ViewSize=<number of bytes>
```

In an XML import file, an 'event' is a single posting on a channel. If the data relating to that 'event' exceeds the specified view size (this includes any metadata, including participants, group name and date), the import file will fail.

Note: The minimum event size is also 250 KB. You cannot set this parameter to a size less then 250 KB.

Important! Do not confuse these 'events' with events in the CA DataMinder database.

More information:

Mapping IM Conversations to Users (see page 97)

Configure IMlogic Dump Files

Note: Importing from IMlogic dump files has been tested using IM Manager 7. IM Import may successfully import dump files generated by other versions of IM Manager, but these have not been tested.

You must configure IMLogic to apply an XSL transform to its dump files. This transform, IMLogicTransform.xsl, is available from CA Technical Support. It ensures that the dump files are converted into a format supported by IMFrontEnd.exe.

To enable IM Import support for IMlogic dump files

1. Launch IMlogic IM Manager.
2. In the Administrator console, set up an export job.
3. On the Manage Account page:

   Message Suffix

   Change to 'xml'.

   XML Output Directory

   This is used for temporary XML files. It can be set to any folder.

   Output directory

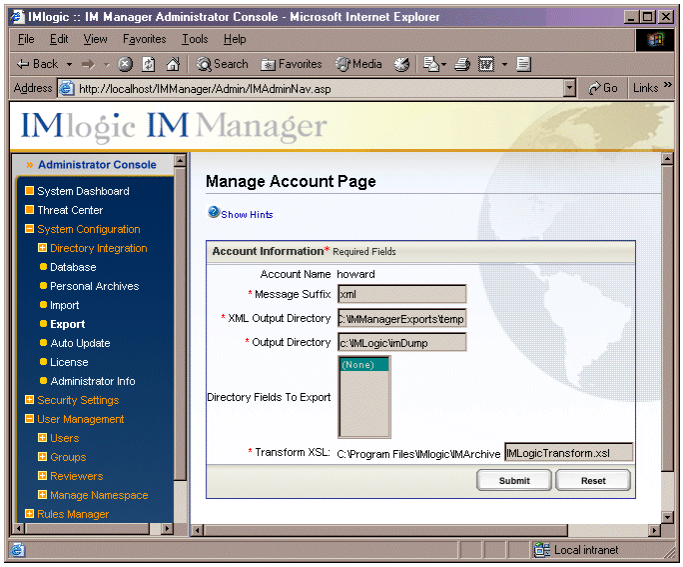
   By default, this is set to the Mail pickup directory. Change this to the IMFrontEnd pickup directory for this as specified by the DirIMlogic parameter.

   Transform XSL

   Set this to IMLogicTransform.xsl.
4. Copy the IMLogicTransform.xsl transform to the archive subfolder in the IMLogic installation folder specified in step 3, typically \IMlogic\IMArchive.

5. Configure IMFrontend.exe to extract and process the IMlogic dump file.

   To do this, when you set up the configuration file IMFrontEnd.ini set the DirIMlogic parameter to the output directory specified in step 3.



IM Manager Administrator console: Manage Account Page

Chapter 5: Third-Party Integration

The following section focuses on CA DataMinder integration with third-party solutions such as email archiving applications. You can also integrate custom archives (see page 116).

In particular, this section describes how emails stored in a third-party archive are still searchable using the iConsole or Data Management console; and how, for some integrations, emails can be categorized (using smart tags) before they are archived.

How archive integration works

1. The archive passes an email (or file) to CA DataMinder for policy analysis.
2. CA DataMinder returns optional SmartTags for the archive to store.
3. The Archive passes a unique ID which CA DataMinder stores in the database with the event.
4. An event BLOB is stored in the 30-day cache of the CMS.
5. If the data is needed after expiry from the cache (for example, because a user accesses an event in the iConsole), the CMS makes a request to the Remote Data Manager. The Remote Data Manager retrieves the email form the archive and stores it back in the 30-day cache.

Integration Components

IBM Content Collector

CA DataMinder can extract emails from IBM Content Collector and apply smart tags to these emails before they are stored in an Enterprise Content Management system such as FileNet P8.

Autonomy ZANTAZ EAS integration

CA DataMinder can extract emails from the EAS archive (using the External Agent API) and import them into the CMS.

Symantec Enterprise Vault integration

CA DataMinder can intercept emails extracted from an Exchange journal mailbox or a Lotus Domino journal and apply smart tags to these emails before they are archived in Enterprise Vault.

EMC SourceOne integration

CA DataMinder can intercept emails extracted from an Exchange journal mailbox or, for Domino, from the EMC Mail Journal Database and apply smart tags to these emails before they are archived in SourceOne.

External Agent API

The External Agent API connects to third party applications and extracts archived emails as EVF files to a cache folder that can be accessed by Event Import.

Socket API

The Socket API, available as an External Agent subfeature, enables external applications and CA DataMinder components such as the NBA and Milter MTA agent to use socket connections to call the External Agent API from a remote location, including from a non-Windows system.

Remote Data Manager (RDM)

The RDM retrieves email data that has been archived in third party remote storage locations. It enables reviewers to retrieve and view archived emails when running event searches in, say, the iConsole and Data Management console.

ICAP Agent

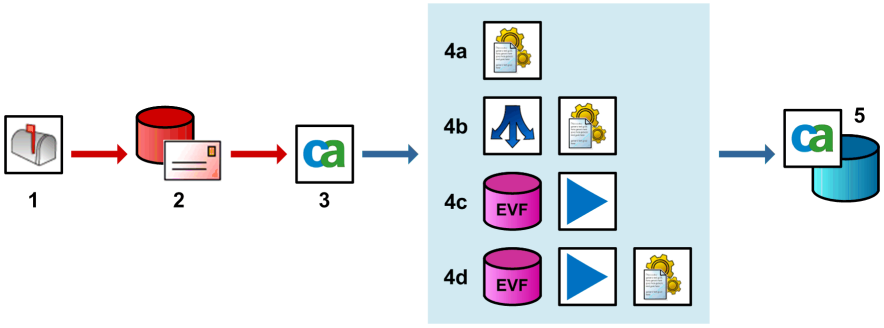
The ICAP Agent enables CA DataMinder to integrate with Internet Content Adaptation Protocol (ICAP) clients running on proxy servers (such as Blue Coat ProxySG). This provides CA DataMinder with a further method for controlling HTTP activity such as file uploads and downloads.

Integration Models

CA DataMinder uses various methods to integrate with third party e-mail archiving products. This is necessary to accommodate the diverse processes used by different archiving products. These integration methods can be grouped into three basic models, summarized below and on the following pages. Note also that each integration model can use one of three methods for ingesting events into CA DataMinder.

Model 1: Push from Archive

The following illustration describes this model:



Integration model 1: Push from archive

In this model, a third party application (step **2** in the diagram below) passes archived emails from a journal mailbox (**1**) to CA DataMinder. At this point in the integration model, there are four methods (**4a** through **4d**) for ingesting the emails into the CMS:

- **4a: Agent outputs to local policy engine:** The archive solution notifies a CA DataMinder integration agent (**3**) that emails are available for processing. The agent then passes the emails to a local policy engine (**4a**). The resulting events are replicated to the CMS (**5**).
- **4b: Agent outputs to policy engines via hub:** As for **4a**, except that the CA DataMinder agent passes the emails to remote policy engines via a PE hub (**4b**).
- **4c: Agent outputs to EVFs for direct import into the CMS:** The archive solution uses the CA DataMinder External Agent API (**3**) to convert the archived emails to event (EVF) files. These EVFs are saved in a cache (**4d**), from where they are subsequently imported directly onto the CMS (**5**).
- **4d: Agent outputs to EVFs to be ingested by Import Policy job:** The archive solution uses the External Agent API (**3**) to convert the archived e-mails to event (EVF) files. These EVFs are saved in a cache (**4c**), from where they are retrieved by Event Import and passed to a local policy engine for processing. The resulting e-mail events are then replicated up to the CMS (**5**).

For details about the operational benefits associated with each ingestion method. In all cases, only the event metadata is saved on the CMS. The actual message data is not stored on the CMS; instead, a database record for each email references the associated entry in the archive.

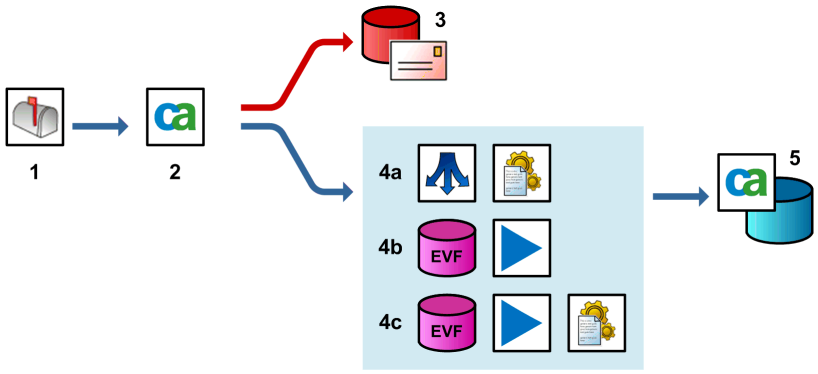
Archive integrations currently using this model are listed beliow.

More information:

Autonomy ZANTAZ EAS Integration (see page 147)
EMC SourceOne Integration (see page 179)
IBM Content Collector (see page 119)
Symantec Enterprise Vault Integration (see page 153)

Model 2: Push to Archive (direct)

The following illustration describes this model:



Integration model 2: Push to archive (direct)

In this model, CA's Universal Adapter (step **2** in the diagrams below) imports e-mails from journal mailboxes (**1**), adds a unique ID to each email and outputs them to an archive adapter (**3**). The archive adapter then outputs them to the archive.

After confirmation that the emails have been successfully archived, the Universal Adapter (UA) then outputs the same e-mails again. At this point in the integration model, there are three methods for ingesting the emails into the CMS:

- **4a: UA outputs to policy engines:** The UA commits the e-mails directly to a policy engine (**4a**). The resulting e-mail events are then replicated up to the CMS (**5**).
- **4b: UA outputs to EVFs to be ingested by Import Policy job:** The UA converts the emails to EVF file. These EVFs are saved in a cache (**4b**), from where they are retrieved by Event Import and passed to a local policy engine for processing. The resulting email events are then replicated up to the CMS (**5**).
- **4c: UA outputs to EVFs for direct import into CMS:** The UA converts the emails to EVF files. These EVFs are saved in a cache (**4c**), from where they are subsequently imported onto the CMS (**5**).

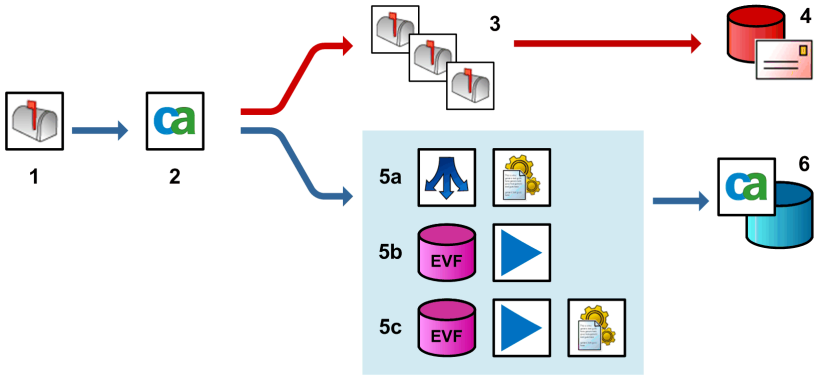
As in model 1, only the event metadata is saved on the CMS. The actual message data is not stored on the CMS; instead, a database record for each email references the associated entry in the archive.

More information:

Universal Adapter (see page 235)

Model 3: Push to Archive (via mailbox)

The following illustration describes this model:



Integration model 3: Push to archive (via mailbox)

In this model, CA's Universal Adapter (step **2** in the diagram below) imports e-mails from journal mailboxes (**1**), adds a unique ID to each email and outputs them to mailboxes (**3**), from where they are picked up by the archive solution (**4**).

After confirmation that the emails have been successfully archived, the Universal Adapter also outputs the same emails (**5**) either to a policy engine or to EVF files. As with models 1 and 2, there are three methods for ingesting the emails into the CMS:

- **5a: UA outputs to policy engines:** The UA commits the e-mails directly to a policy engine (**5a**). The resulting e-mail events are then replicated up to the CMS (**6**). Unlike ingestion methods 5b and 5c, this method has no additional storage overhead.
- **5b: UA outputs to EVFs to be ingested by Import Policy job:** The UA converts the emails to EVF file. These EVFs are saved in a cache (**5b**), from where they are retrieved by Event Import and passed to a local policy engine for processing. The resulting e-mail events are then replicated up to the CMS (**6**). Like method 5a, this method allows smart tags to be assigned to the ingested e-mails, but also permits ingestion to be scheduled for periods of low network activity.

- **5c: UA outputs to EVFs for direct import into CMS:** The UA converts the e-mails to EVF files. These EVFs are saved in a cache (**5c**), from where they are subsequently imported onto the CMS (**6**).

As in models 1 and 2, only the event metadata is saved on the CMS. The actual message data is not stored on the CMS; instead, a database record for each email references the associated entry in the archive.

More information:

Universal Adapter (see page 235)

Comparison of Ingestion Methods into CA DataMinder

Each archive integration model supports three methods for ingesting e-mails into the CMS (methods a, b, and c). In each case, CA DataMinder (either an integration agent or the Universal Adapter) processes the emails to be ingested and outputs them to EVF files or to policy engines. Each method has its own benefits, and you must choose the method most appropriate for your organization:

	Ingestion method		
Operational benefit	**a.**	**b.**	**c.**
Smart tags stored with emails in archive	Yes		
Smart tags stored with email events in CMS	Yes	Yes	
Decouples archiving from ingestion		Yes	Yes
Potentially high ingestion rates			Yes

a. Output to Policy Engines

The integration agent or UA outputs emails using a policy engine hub to a policy engine. Unlike the EVF-based ingestion methods, ingestion method a. has no additional storage overhead. This method has the following benefits:

- **Smart tags stored with emails in archive:** It allows you, using policy engines, to add smart tags to emails **before** they are stored in your archive.
- The UA can automatically assign smart tags to emails before they are archived. But if your archive integration uses the External Agent API, your own integration software must handle any smart tags returned from the External Agent, saving them as properties of the emails before they are archived.
- **Smart tags stored with email events in CMS:** You can also store smart tags with the corresponding email events on the CMS. This allows rapid and efficient filtering when using CA DataMinder consoles to search for archived events.

b. Output to EVFs, then Ingested by Import Policy

The integration agent or UA converts archived emails to event (EVF) files. These EVFs are saved in a cache, from where they are retrieved by Event Import and passed to a policy engine for processing. This method has the following benefits:

- **Smart tags stored with email events in CMS:** As with method a, you can store smart tags with email events when they are saved on the CMS.
- **Decouples archiving from ingestion:** Saving emails in your archive and ingesting the corresponding events into CA DataMinder are performed as two separate processes. This lets you archive emails as soon as possible while scheduling ingestion for periods of low network activity. This is particularly useful if complex policy processing is required.

  However, because ingestion method b. relies on an interim EVF cache you must add fault-tolerance to the cache by implementing regular backups.

c. Output to EVFs, then Ingested Directly

The integration agent or UA converts archived emails to event (EVF) files. These EVFs are saved in a cache, from where they are subsequently imported directly onto the CMS by Event Import. This method has these benefits:

- **High ingestion rates:** Because no policy processing is required, this ingestion method is potentially the fastest.
- **Decouples archiving from ingestion:** As with method b, archiving and ingestion into CA DataMinder are two separate processes. This allows you to schedule ingestion for periods of low network activity.

  However, ingestion method c. does not allow smart tags to be assigned to the ingested emails. And like method b, you must add fault-tolerance to the cache by implementing regular backups.

Supported Archive Versions

CA DataMinder integrates with the following versions of these products:

Autonomy ZANTAZ EAS

The supported versions of ZANTAZ EAS are:

- Version 4.x
- Version 5 SP1

Note: We expect the current version of CA DataMinder to integrate successfully with these versions of Autonomy ZANTAZ EAS, but this integration has not been tested.

EMC SourceOne Email Management for Exchange or Domino

CA DataMinder supports SourceOne 6.6 SP1 or later. But see the note.

Note: CA DataMinder does not support SourceOne 6.8.2 due to incompatible product changes introduced in this version.

IBM Content Collector

CA DataMinder supports Content Collector V3.0. It may support other versions, but these have not been tested.

Symantec Enterprise Vault

CA DataMinder supports Enterprise Vault 9.0 SP1 or higher for Exchange and Domino emails.

Custom Archive Integration

In addition to the explicitly supported archives, CA DataMinder lets 3rd-party integrators develop support for custom archives.

The integrator must do the following to support the archive:

1. Develop a component responsible for receiving new items from the archive, and passing them to CA DataMinder's External Agent API for policy analysis.
2. Develop a component responsible for retrieving historic items from the archive when requested by the Remote Data Manager.

More information:

Support for Custom Archives (see page 196)
Remote Data Manager Support for Custom Archives (see page 224)

Support for Regional Archives

CA DataMinder supports multiple unconnected email archive instances on a single Central Management Server. If your enterprise has separate installations of the same email archive in multiple geographic regions, it is now possible to segregate the events from these onto a single CMS so that the Remote Data Manager can retrieve historic emails from the correct archive installation.

To configure a region:

1. Specify the RegionTag registry setting (see page 193) on all machines to which the archive sends emails for analysis, that is, the archive agent machines.
2. Dedicate one or more Remote Data Manager servers to each region. (see page 233) They are responsible for retrieving emails for that region from the archive.

Chapter 6: IBM Content Collector

This section contains the following topics:

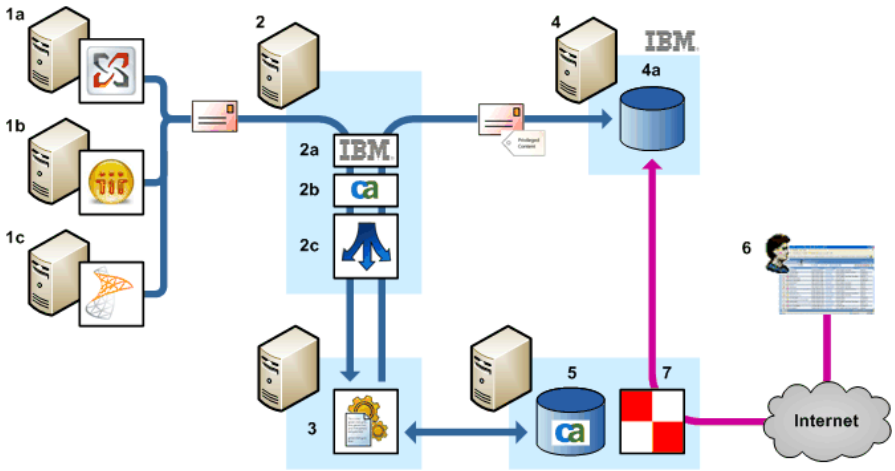
Overview

CA DataMinder can integrate with IBM Content Collector. Integration is provided through custom task routes and a set of custom metadata.

How does the integration work? Content Collector collects emails from an email source. The source can be an Exchange, Domino or SMTP server. The task route then forwards these emails to the CA DataMinder archive agent for Content Collector. The archive agent then passes the emails to CA DataMinder policy engines. The policy engines analyze the emails and apply user policy. Specifically, you configure CA DataMinder user policy to apply smart tags to these emails. For example, you can use smart tags to indicate archive retention periods.

When policy analysis is complete, the emails plus any resulting smart tags are returned to Content Collector for storage in the archive.

The key components are illustrated in the diagram below. For simplicity, this diagram shows Content Collector passing emails to a single CA DataMinder policy engine. In practice, large organizations may use multiple policy engines.



CA DataMinder integration with IBM Content Collector

1. **Email source.** Content Collector collects emails from an email source. This can be an Exchange journal mailbox (**1a**), a Domino journal database (**1b**), or a mailbox on an SMTP server (**1c**).
2. **IBM Content Collector.** Content Collector (**2a**) collects emails from the source and routes to the Content Collector archive agent (**2b**). In turn, the archive agent passes emails to a policy engine hub (**2c**). The hub then distributes emails to policy engines (**3**).
3. **Policy engine**. The policy engine analyzes the emails, applies policy triggers, and attaches smart tags as necessary.
4. **Enterprise Content Management (ECM) system.** For this example, the ECM is IBM FileNet P8. Processed emails plus any resulting smart tags are returned to P8 (**4a**) for archiving.
5. **CMS.** Events generated as a result of policy processing, including metadata that incorporates the unique message identifiers, are replicated to the CMS and stored in the database. In this example, the CMS machine also hosts the Remote Data Manager (**7**).
6. **iConsole.** Reviewers can search for archived emails.
7. **Remote Data Manager (RDM).** When a user searches for archived emails in the iConsole (**6**), the RDM retrieves data for these emails. In this example, the RDM retrieves emails from P8 (**4a**) via Content Collector.

Integration Requirements

Note the following requirements for integration with IBM Content Collector.

IBM Content Collector

CA DataMinder supports Content Collector V3.0. It may support other versions, but these have not been tested.

Operating System

The server hosting the Content Collector archive agent must be running Windows Server 2008 R2 or later.

Integration Procedure for IBM Content Collector

Integrating CA DataMinder with Content Collector involves the following tasks:

1. Install the Content Collector archive agent on your IBM Content Collector host server.
2. Configure IBM Content Collector.
   a. Add the required user-defined metadata.
   b. Set up the CA DataMinder task routes.
3. Allow the CA DataMinder Import Web Service to access archived attachments.
4. (Optional) Configure the CA DataMinder web service to use SSL.
5. Deploy your CA DataMinder policy engines.
6. Add CA DataMinder smart tags to Content Collector Emails.
7. Use the Remote Data Manager (RDM) to retrieve archived emails from Content Collector.

More information:

Install the Content Collector Archive Agent (see page 121)
Configure IBM Content Collector (see page 122)
Grant Read Access to the CA DataMinder Import Web Service (see page 139)
DLP--Configure the CA DataMinder Import Web Service to Use SSL (see page 140)
Deploy Policy Engines (see page 141)
Add CA DataMinder Smart Tags to Content Collector Filtering Rules (see page 141)
Use the RDM to Retrieve Content Collector Archived Events (see page 144)

Install the Content Collector Archive Agent

You need to install the Content Collector archive agent on your Content Collector server. A policy engine hub is installed automatically with the Content Collector archive agent.

To install the Content Collector Archive 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 Custom Setup screen.

5. In the Custom Setup screen, expand the Archive Agents feature and select IBM Content Collector.

   A policy engine hub is installed automatically with this agent.

6. In the Policy Engine Selection screen, identify the policy engine host server and provide the credentials for the PE domain user.

   - Enter name or IP address of PE host server
   - In Windows Credentials section, enter the name and password of a domain administrator. The PE hub uses this account to connect to the policy engine.

7. In the final wizard screen, click Install to start the file transfer.

Configure IBM Content Collector

After installing the Content Collector archive agent, you must configure Content Collector to enable Task Routes to call the CA DataMinder web service API. The steps are summarized below:

1. Add User-Defined Metadata for Web Service Requests and Responses.

   You must add user-defined metadata entities for all data sent in web service requests and returned by web service responses. Specifically, you must create user-defined metadata entities to define the web service request and response parameters.

   Specifically, you must add the following metadata entities:

   - CA DataMinder Import Analyse Request Metadata
   - CA DataMinder Import Commit Request Metadata
   - CA DataMinder Import Metadata Response Metadata

   For each metadata entity, you must define its properties and GUID.

2. Set up the CA DataMinder Task Routes

   The Content Collector archive agent includes three task route templates for integrating CA DataMinder into your email collection task routes. You must create new task routes based on these templates and apply the metadata that you added in step 1 to these task routes.

   Note: The three Task Route templates collect emails from Microsoft Exchange IBM Lotus Domino, Microsoft Exchange, and SMTP messaging systems.

   For each task route, you must:

   a. Edit the properties of the CA DataMinder Analyze Request task.

      In particular, you must define the property mappings for this task.

   b. Edit the properties of the subsequent Audit Log task.

   c. Edit the properties of the CA DataMinder Commit Request task.

   d. Edit the properties of the subsequent Audit Log task.

   e. In the Error task route, edit the properties of the CA DataMinder Rollback Request task.

   f. Configure the task route collector.

   g. Activate the task route.

More information:

Add User-Defined Metadata for Web Service Requests and Responses (see page 124)
Set up CA DataMinder Task Routes (see page 131)

Add User-Defined Metadata for Web Service Requests and Responses

You must add user-defined metadata for all data sent in CA DataMinder web service requests and returned in web service responses.

Note: The following instructions refer to IBM Content Collector V3.0.

To add 'Analyze Request' metadata

1. Launch the IBM Content Collector Configuration Manager
2. In the Metadata and Lists tab, click 'User-Defined Metadata'.
3. In the Configured User-defined Metadata pane, click the Add button.

   This creates a User-defined Metadata pane, comprising two sections: General and Metadata Properties.
4. In the General section, type the name of the metadata entity in the Display Name field:

   `CA DataMinder Import Analyze Request Metadata`

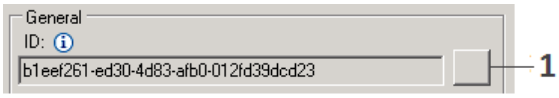
   Important! Enter the name exactly as shown above. This name will be used to define the JSON used in the web service request and response.
5. In the Metadata Properties section, click the Import Metadata Properties button.

   The Import Metadata Properties dialog appears.

   a. In the File Name field, click the browse button and select this file:
   `CA DataMinder Import Analyze Request Metadata.xml`

   Find this file in the \CA\CA DataMinder\ICC subfolder below the *64-bit* CA DataMinder installation folder.

   b. Click OK to import the required properties for this metadata. These properties are listed in the following section (see page 127).
6. In the General section, change the system-defined GUID in the ID field:



   To do this, click the square Edit button (**1**) to right of the ID field. Then replace the default GUID with:

   `ca.dataminder.metadata.import.analyze.request`

   Important: Enter the new GUID exactly as shown above. Content Collector uses this ID value to map the response returned from the CA DataMinder web service.
7. Click File, Save to save the new metadata.

   Now add the Commit Request metadata.

To add 'Commit Request' metadata

1. Launch the IBM Content Collector Configuration Manager
2. In the Metadata and Lists tab, click ‘User-Defined Metadata’.
3. In the Configured User-defined Metadata pane, click the  Add button.
4. In the General section of the User-defined Metadata pane, type the following name in the Display Name field:
   `CA DataMinder Import Commit Request Metadata`

   Important! Enter the name exactly as shown above.
5. In the Metadata Properties section, click the  Import Metadata Properties button.

   The Import Metadata Properties dialog appears.

   a. In the File Name field, click the  browse button and select this file:
      `CA DataMinder Import Commit Request Metadata.xml`

      Find this file in the \CA\CA DataMinder\ICC subfolder below the *64-bit* CA DataMinder installation folder.

   b. Click OK to import the required properties for this metadata. These properties are listed in the following section (see page 129).
6. In the General section, change the system-defined GUID in the ID field. To do this, click the square Edit button to right of the ID field. Then replace the default GUID with:
   `ca.dataminder.metadata.import.commit.request`

   Important: Enter the new GUID exactly as shown above.
7. Click File, Save to save the new metadata.

   Now add the Response metadata.

To add 'Response' metadata

1. Launch the IBM Content Collector Configuration Manager
2. In the Metadata and Lists tab, click 'User-Defined Metadata'.
3. In the Configured User-defined Metadata pane, click the  Add button.
4. In the General section of the User-defined Metadata pane, Type the following name in the Display Name field:
   `CA DataMinder Import Metadata Response`

   Important! Enter the name exactly as shown above.
5. In the Metadata Properties section, click the  Import Metadata Properties button.

   The Import Metadata Properties dialog appears.

   a. In the File Name field, click the  browse button and select this file:
      `CA DataMinder Import Metadata Response.xml`

      Find this file in the \CA\CA DataMinder\ICC subfolder below the *64-bit* CA DataMinder installation folder.

   b. Click OK to import the required default properties for this metadata. These default properties for are listed in the following section (see page 130).

   Note: When you define your CA DataMinder smart tags (see page 142), you must add corresponding properties for each smart tags to this list in the Metadata Properties section.
6. In the General section, change the system-defined GUID in the ID field. To do this, click the square Edit button to right of the ID field. Then replace the default GUID with:
   `ca.dataminder.import.response`

   Important: Enter the new GUID exactly as shown above.
7. Click File, Save to save the new metadata.

More information:

Properties: CA DataMinder Import Analyse Request Metadata (see page 127)
Properties: CA DataMinder Import Commit Request Metadata (see page 129)
Properties: CA DataMinder Import Metadata Response (see page 130)

Properties: CA DataMinder Import Analyse Request Metadata

This section describes the properties required for CA DataMinder Import Analyse Request Metadata.

When you import the associated XML file, the following properties are imported into the Metadata Properties section of the User-defined Metadata pane:

Display Name: AttachmentFile

Property ID: AttachmentFile

Data type: String

Display Name: AttachmentName

Property ID: AttachmentName

Data type: String

Display Name: BCCRecipients

Property ID: BCCRecipients

Data type: StringArray

Display Name: CCRecipients

Property ID: CCRecipients

Data type: StringArray

Display Name: DocumentTitle

Property ID: DocumentTitle

Data type: String

Display Name: IsAttachment

Property ID: IsAttachment

Data type: Boolean

Display Name: IsEncrypted

Property ID: IsEncrypted

Data type: Boolean

Display Name: JournalHeaders

Property ID: JournalHeaders

Data type: String

Display Name: MessageBody

Property ID: MessageBody

Data type: String

Display Name: MessageFrom

Property ID: MessageFrom

Data type: String

Display Name: MessageSize

Property ID: MessageSize

Data type: Integer

Display Name: OriginatingUser

Property ID: OriginatingUser

Data type: String

Display Name: ReceivedDate

Property ID: ReceivedDate

Data type: DateTime

Display Name: Recipients

Property ID: Recipients

Data type: StringArray

Display Name: SentDate

Property ID: SentDate

Data type: DateTime

Display Name: Subject

Property ID: Subject

Data type: String

Display Name: ToAddress

Property ID: ToAddress

Data type: StringArray

Display Name: UniqueID

Property ID: UniqueID

Data type: String

Properties: CA DataMinder Import Commit Request Metadata

This section describes the properties required for CA DataMinder Import Commit Request Metadata.

When you import the associated XML file, the following properties are imported into the Metadata Properties section of the User-defined Metadata pane:

Display Name: ObjectID

Property ID: ObjectID

Data type: String

Display Name: Repository Name

Property ID: RepositoryName

Data type: String

Properties: CA DataMinder Import Metadata Response

This section describes the properties required for CA DataMinder Import Commit Request Metadata.

Default Properties

When you import the associated XML file, the following properties are imported into the Metadata Properties section of the User-defined Metadata pane:

Display Name: CA DataMinder Status Code

Property ID: CADataMinderStatusRCode

Data type: String

Display Name: CA DataMinder Status Message

Property ID: CADataMinderStatusMessage

Data type: String

Display Name: IsAnalyze

Property ID: IsAnalyze

Data type: Boolean

Display Name: IsCommit

Property ID: IsCommit

Data type: Boolean

Smart Tag Properties

You must also add a custom property for each smart tag defined in your CA DataMinder user policies.

Display Name: <smart tag>

Property ID: <smart tag name>

Data type: String

For example, if your CA DataMinder user policies apply a smart tag named 'Privileged_Content', configure the metadata property as follows:

Display Name: Privileged_Content

Property ID: Privileged_Content

Data type: String

Set up CA DataMinder Task Routes

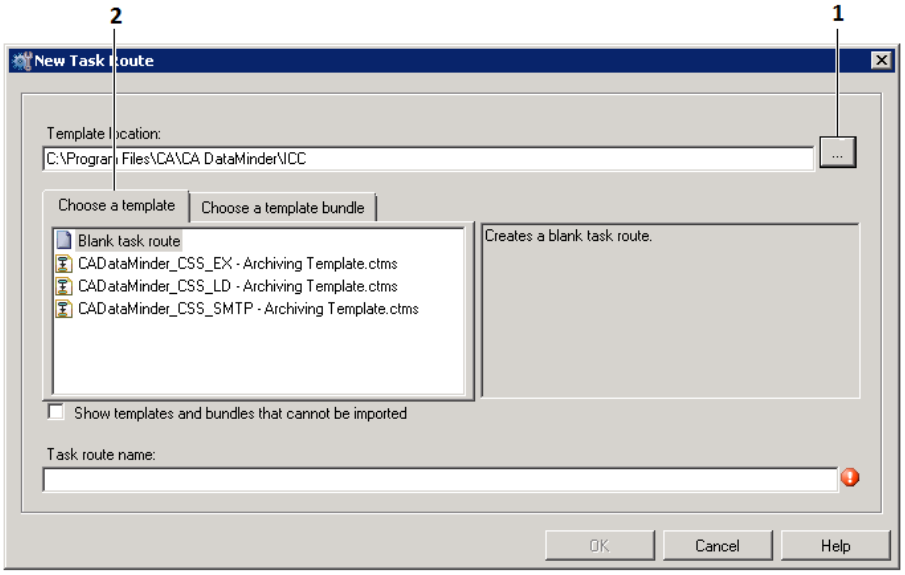
After creating the user-defined metadata that is needed to integrate Content Collector with CA DataMinder, you must apply that metadata to CA DataMinder task route(s).

To apply user-defined metadata to CA DataMinder task routes

Note: The following instructions and screenshots refer to IBM Content Collector V3.0.

1. Launch the IBM Content Collector Configuration Manager.
2. Click the Task Routes button to open the Task Routes pane
3. Click the  New button to create a new CA DataMinder task route. You must base the new task on one of the CA DataMinder task route templates.

   The New Task Route dialog displays:



   1 Browse button. **2** Choose a Template tab.

4. Click the browse button to the right of the Template location field.

   The Browse For Folder dialog appears.

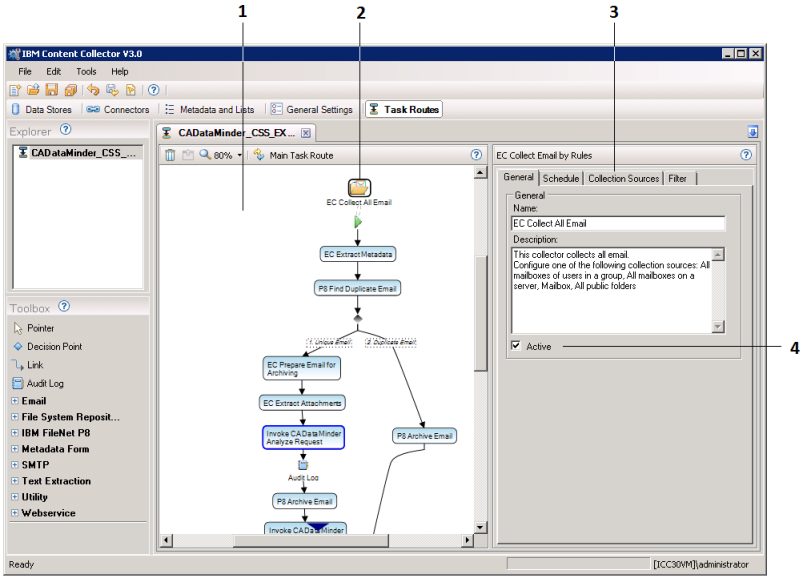
5. Browse to the \CA\CA DataMinder\ICC subfolder below the *64-bit* CA DataMinder installation folder and click OK to close the dialog

   The CA DataMinder task route templates are listed in the Choose a Template tab of the New Task Route dialog.

6. Select a task route template and click OK.

   For example, select the template for Exchange emails:
   `CA DataMinder_CSS_EX_Archiving Template.ctms`

The new task route displays:



Task Route screen: **1** Individual tasks in task route. **2** Task collector. **3** Task properties. **4** Activate check box.

7. Click the following task to view its properties in the right-hand pane.
   `Invoke CA DataMinder Analyze Request`

   Edit the task properties. For details, see the following section (see page 134).

8. Click the following task to view its properties in the right-hand pane:
   `Audit Log`

   Edit the task properties. For details, see the following section (see page 136).

   Note: This is the Audit Log task that immediately follows the CA DataMinder Import Analyze Response Metadata task. It is the first of two Audit Log tasks.

9. Click the following task to view its properties in the right-hand pane:
   `Invoke CA DataMinder Commit Request`

   Edit the task properties. For details, see the following section (see page 134).

10. Click the following task to view its properties in the right-hand pane:
    `Audit Log`

    Edit the task properties. For details, see the following section (see page 138).

    Note: This is the Audit Log task that immediately follows the CA DataMinder Import Commit Response Metadata task. It is the second of two Audit Log tasks.

11. Switch to the Error Task Route and click the following task to view its properties in the right-hand pane:
    `Invoke CA DataMinder Rollback Request task`

    Edit the task properties. For details, see the following section (see page 138).

12. Now configure the task collector ('SC Collect All Email'; see the previous screenshot). Click the collector to view its properties in the right-hand pane.

    In the Schedule tab, specify how often the task runs.

    In the Collection Sources tab, specify the email source, typically a Journal or Mailbox. For example, enter a mailbox SMTP address such as sales@unipraxis.com.

13. Click File, Save All to save the new task route.

14. Now activate the task route. Click the collector to view its properties in the right-hand pane.

    In the General tab, select the Activate check box to activate the task (see the previous screenshot).

15. Confirm that the IBM Content Collector Task Routing Engine service has started on the host server.

More information:

Configure Task: CA DataMinder Analyze Request (see page 134)
Configure Task: Audit Log #1 (see page 136)
Configure Task: CA DataMinder Commit Request (see page 137)
Configure Task: Audit Log #2 (see page 138)
Configure Task: CA DataMinder Rollback Request (see page 138)

Configure Task: CA DataMinder Analyze Request

Now edit the properties of the Invoke CA DataMinder Analyze Request task.

To edit the task properties

1. Open a CA DataMinder task route (see page 131) in the IBM Content Collector Configuration Manager.
2. Click the following task to view its properties in the right-hand pane:
   `Invoke CA DataMinder Analyse Request`
3. Edit the task properties in the following sections:

   Connection Details

   The Import web service is always installed on the Content Collector host server, so you do not need to change the default connection URL:
   `http://localhost/ICCImportSvc/import/analyze`

   Webservice response mapping

   Set the Metadata Produced By Service field to:
   `CA DataMinder Import Metadata Response`

   Property Mappings

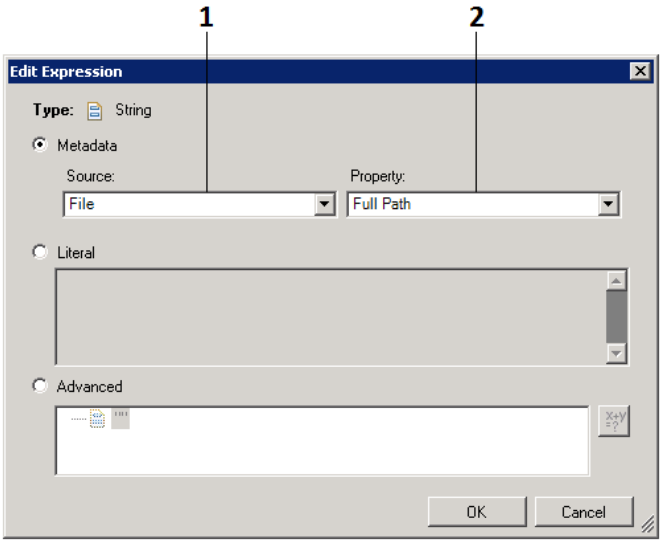
   These properties define which metadata is sent to the CA DataMinder web service. Set the Class field to:
   `CA DataMinder Analyze Metadata Request`

   All the properties that you defined in the user-defined metadata section now appear in the Property column. But the Value column remains blank. You must now populate the Value column with source-property mappings for each task property.
4. Double-click a property row in the Property Mappings table.

   The Edit Expression dialog appears.
5. Click the Metadata option.
6. Define the Source-Property mapping for the property that you are editing:
   - In the Source field, choose the source class. This is always either File or Email.
   - In the Property field, choose the correct item from the list. Refer to the table below to find the correct property.

For example, for the AttachmentFile property you must map the the File class to FullPath. This gives the following mapping:

<File, FullPath>



Edit Expression dialog

7. Click OK to close the Edit Expression dialog and return to the main Task Route screen.
8. Repeat steps 4 though 7 for each task property.

   The complete list of property mappings for the CA DataMinder Analyze Request task are listed below.
9. Click Save to save your changes.

   Important! A known issue causes the values to disappear in the Property Mappings table. Do not worry about this. The new values have been saved correctly.

Property Mappings for CA DataMinder Analyze Request Task

Property	Value	Datatype
AttachmentFile	<File, FullPath>	String
AttachmentName	<File, FullPath>	String
BCCRecipients	<Email,BCCAddress(multi)>	Multi String
CCRecipients	<Email,CCAddress(multi)>	Multi String
DocumentTitle	<Email,DocumentTitle>	String
IsAttachment	<Email,Is Attachment>	Boolean
IsEncrypted	<Email, IsEncrypted>	Boolean

Property	Value	Datatype
JournalHeaders	<Email, JournalEnvelopeHeaders>	String
MessageBody	<Email, MessageBody>	String
MessageForm	<Email, MessageForm>	String
MessageSize	<Email, MessageSize>	Int
OriginatingUser	<Email, OriginatingUser>	String
ReceivedDate	<Email, ReceivedDate>	Date Time
Recipients	<Email,RecipientsAddresses(multi)>	Multi String
SentDate	<Email,SentDate>	Date Time
Subject	<Email,Subject>	String
ToAddress	<Email,ToAddress (multi)>	Multi String
UniqueId	<Email,UniqueId>	String

Configure Task: Audit Log #1

Now edit the properties of the first Audit Log task.

In the Audit Log Fields list, include the CA DataMinder Import Analyze Request Metadata.

Configure Task: CA DataMinder Commit Request

Now edit the properties of the CA DataMinder Commit Request task.

To edit the task properties

1. Open a CA DataMinder task route (see page 131) in the IBM Content Collector Configuration Manager.
2. Click the following task to view its properties in the right-hand pane:
   `Invoke CA DataMinder Commit Request`
3. Edit the task properties in the following sections:

   Connection Details

   The Import web service is always installed on the Content Collector host server, so you do not need to change the default connection URL:
   `http://localhost/ICCImportSvc/import/commit`

   Webservice response mapping

   Set the Metadata Produced By Service field to:
   `CA DataMinder Commit Metadata Request`

   Property Mappings

   These properties define which metadata is sent to the CA DataMinder web service. Set the Class field to:
   `CA DataMinder Commit Metadata Request`

   All the properties that you defined in the user-defined metadata section now appear in the Property column. But the Value column remains blank. You must now populate the Value column with source-property mappings for each task property.
4. Double-click a property row in the Property Mappings table.

   The Edit Expression dialog appears.
5. Click the Metadata option.
6. Define the Source-Property mapping for the property that you are editing:
   - In the Source field, choose the source class. This is always either File or Email.
   - In the Property field, choose the correct item from the list. Refer to the table below to find the correct property.

   For example, for the ObjectID property you must map the the P8 Create Document source to ObjectID. This gives the following mapping:
   `<P8 Create Document, ObjectID>`
7. Click OK to close the Edit Expression dialog and return to the main Task Route screen.
8. Repeat steps 4 through 7 for each task property.

The complete list of property mappings for the CA DataMinder Commit Request task are listed below.

Property Mappings for CA DataMinder Commit Request Task

Property	Value	Datatype
Object ID	<P8 Create Document, ObjectID>	String
Repository	<P8 Create Document, ConnectionName>	String

Configure Task: Audit Log #2

Now edit the properties of the second Audit Log task.

In the Audit Log Fields list, include the CA DataMinder Import Commit Request Metadata.

Configure Task: CA DataMinder Rollback Request

Now edit the properties of the CA DataMinder Rollback Request task.

To edit the task properties

1. Open a CA DataMinder task route (see page 131) in the IBM Content Collector Configuration Manager.
2. Switch to the Error task route.
3. Click the following task to view its properties in the right-hand pane:
   `Invoke CA DataMinder Rollback Request task`
4. Edit the task properties in the following sections:

   Connection Details

   The Import web service is always installed on the Content Collector host server, so you do not need to change the default connection URL:
   `http://localhost/ICCImportSvc/import/rollback`

   Webservice response mapping

   Set the Metadata Produced By Service field to:
   `CA DataMinder Import Metadata Response`
5. No further setup is required for this task. Switch back to the Main Task Route and continue the task route setup.

Grant Read Access to the CA DataMinder Import Web Service

The CA DataMinder Import Web Service requires read access to the folder that IBM Content Collector uses to store email attachments.

The CA DataMinder Import Web Service runs as the 'NETWORK SERVICE' account and requires Read access to the folder that IBM Content Collector uses to store email attachments. This folder is specified by the Working Directory setting on the properties page of the specified Connector.

Follow these steps:

1. On the IBM Content Collector host server, browse to the Working Directory folder. By default, this folder is:
   `C:\Users\iccadm\AppData\Local\Temp`
2. View the Security properties for this folder.
3. Add the 'NETWORK SERVICE' account to this folder and allow the Read permission.

   Note: Do not ignore the space! Do not try to add a 'NetworkService' account.

DLP--Configure the CA DataMinder Import Web Service to Use SSL

if you require secure communications between Content Collector and the CA DataMinder policy engines, you can configure the CA DataMinder web service to use SSL.

To configure the CA DataMinder web service to use SSL

1. Open the web.config.

   Find this file in the \bin subfolder of the CA DataMinder installation folder on the IBM Content Collector host server.

2. Locate the *<service>* element:

```
<services>
  <service name="ICCImportSvc.WgnICCImportSvc">
    <endpoint binding="webHttpBinding"
contract="ICCImportSvc.IWgnICCImportSvc"/>
  </service>
</services>
```

3. Add the *bindingConfiguration* property to the *<service>* element:

```
<services>
  <service name="ICCImportSvc.WgnICCImportSvc">
    <endpoint
     binding="webHttpBinding"
     contract="ICCImportSvc.IWgnICCImportSvc"
     bindingConfiguration="webHttps"/>
  </service>
</services>
```

4. Add a new *<bindings>* element at the same level as the *<services>* element:

```
<bindings>
  <webHttpBinding>
    <binding name="webHttps">
      <security mode="Transport" />
    </binding>
  </webHttpBinding>
</bindings>
```

   The updated web.config file now looks like this:

```
<services>
  <service name="ICCImportSvc.WgnICCImportSvc">
    <endpoint
     binding="webHttpBinding"
     contract="ICCImportSvc.IWgnICCImportSvc"
     bindingConfiguration="webHttps"/>
  </service>
</services>
<bindings>
  <webHttpBinding>
    <binding name="webHttps">
```

```
        <security mode="Transport" />
      </binding>
    </webHttpBinding>
  </bindings>
```

5. Verify that SSL is configured in Microsoft IIS.

   The CA DataMinder import web service will accept requests using HTTPS.

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.

Add CA DataMinder Smart Tags to Content Collector Filtering Rules

You now need to define smart tags in your CA DataMinder user policies. When appropriate, policy engines apply these smart tags to emails received from Content Collector.

You must also edit the Content Collector metadata to recognize the smart tags attached to emails by CA DataMinder.

Finally, you must configure the CA DataMinder task routes to use the smart tags. See the following sections for details.

More information:

Add Smart Tags to CA DataMinder User Policies (see page 142)
Add Smart Tag Recognition to Content Collector Metadata (see page 143)
Add Smart Tags to CA DataMinder Task Routes (see page 143)

Add Smart Tags to CA DataMinder User Policies

Follow these steps:

1. From the Administration console, launch the User Policy Editor.

   For details, see the Editing Policies in the Administration Console chapter in the *Policy Guide*.

2. In the User Policy Editor, select the trigger you want to configure.

3. Expand the trigger folder, then right-click the Smart Tags setting and choose Properties.

4. In the Smart Tags Properties dialog, click Add and specify a Smart Tag Name and, optionally, a Smart Tag Value.

   For example, set the smart tag name to:
   `Privileged_Content`

5. Save the policy changes.

   Note: For full details about smart tags, see the Smart Tagging chapter in the *Policy Guide*.

Add Smart Tag Recognition to Content Collector Metadata

Follow these steps:

1. Launch the IBM Content Collector Configuration Manager
2. In the Metadata and Lists tab, click 'User-Defined Metadata'.
3. In the User-Defined Metadata pane, click the following:
   `CA DataMinder Import Metadata Response`

   This displays the properties pane. The properties pane comprises two sections: General and Metadata Properties.
4. In the Metadata Properties section, click the Add button.
5. Add a custom property for each smart tag defined in your CA DataMinder user policies.

   Display Name: <smart tag>

   Property ID: <smart tag name>

   Data type: String

   For example, if your CA DataMinder user policies apply a smart tag named 'Privileged_Content', configure the metadata property as follows:

   Display Name: Privileged_Content

   Property ID: Privileged_Content

   Data type: String

   Note: The Display Name and Property ID must *exactly match* the smart tag in the CA DataMinder user policy.

Add Smart Tags to CA DataMinder Task Routes

Follow these steps:

1. Launch the IBM Content Collector Configuration Manager.
2. Click the Task Routes button to open the Task Routes pane
3. Load the CA DataMinder task route that you want to configure.

   For example, load CADataMinder_CSS_EX - Archiving Template.ctms.
4. Configure the task route to use the required smart tags.

   See your Content Collector documentation for details.

Use the RDM to Retrieve Content Collector Archived Events

When IBM Content Collector passes emails to the CA DataMinder archive agent, each email includes a unique identifier. In turn, the archive agent passes this identifier to a policy engine.

When the policy engine processes the email from Content Collector, it generates a CA DataMinder email event and adds the unique identifier to the event's metadata. The event and its identifier are then replicated to the CMS. The RDM uses this identifier to retrieve the email from the Content Collector archive during subsequent event searches.

To set up event retrieval from the CA DataMinder iConsole, you must install and configure the RDM. Then you must edit the DocViewer.config file on the Content Collector host server.

To install and configure the RDM

1. Install the RDM (see page 225).

   You can install the RDM on any CA DataMinder server. The architecture diagram in the Overview (see page 119) shows the RDM installed on the CMS server.

2. On the RDM host server, specify the Content Collector host server (see page 229).

To edit DocViewer.config

1. On the Content Collector host server, locate the default EmailTemplateCA.XSL file.

   Find this file in the \CA\CA DataMinder\ICC subfolder below the CA DataMinder installation folder.

2. Copy EmailTemplateCA.XSL to this target folder:
   `\ibm\contentcollector\afuweb\docviewer\config`

3. In the target folder, edit the docviewer.config file to reference the XSL file that you copied in step 2. Specifically, do one of the following:

   - Change all 'EmailTemplate.XSL' references to 'EmailTemplateCA.XSL'.
   - (Recommended) Comment out 'EmailTemplate.XSL' references and add new references to 'EmailTemplateCA.XSL'. For example:

     ```
     #FORMTYPE.Memo.0 EmailTemplate.xsl
     FORMTYPE.Memo.0 EmailTemplateCA.xsl

     #MESSAGECLASS.IPM.Note.0 EmailTemplate.xsl
     MESSAGECLASS.IPM.Note.0 EmailTemplateCA.xsl

     #XSLT.DEFAULT.0 EmailTemplate.xsl
     XSLT.DEFAULT.0 EmailTemplateCA.xsl
     ```

4. Save the changes to docviewer.config.

More information:

EMC SourceOne Integration (see page 231)

Chapter 7: Autonomy ZANTAZ EAS Integration

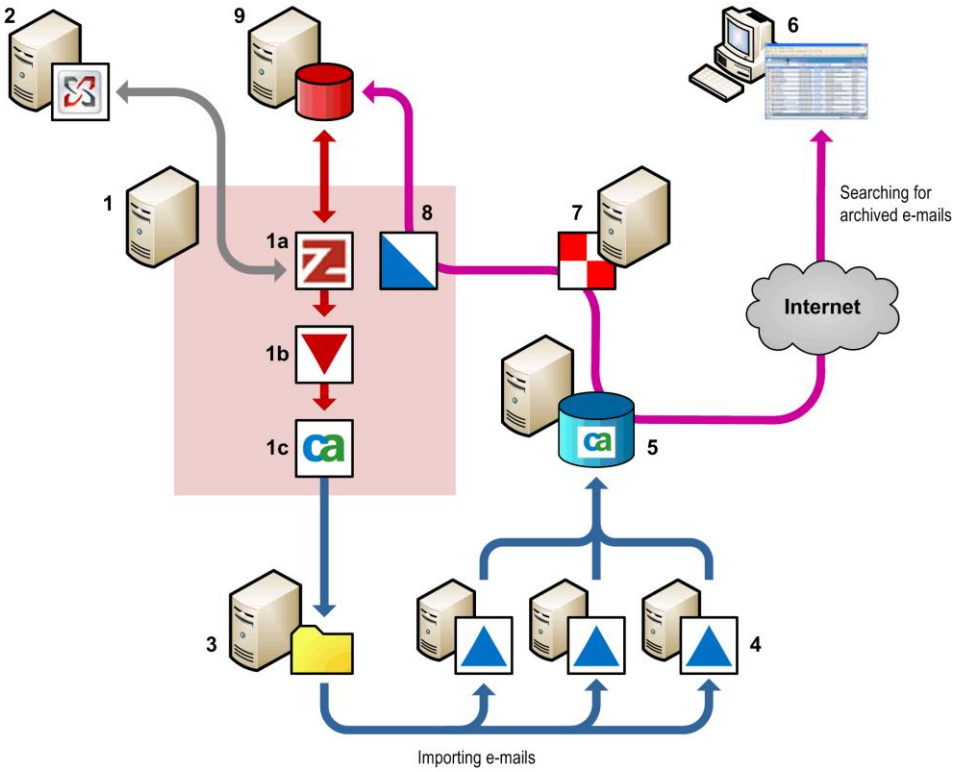
This section contains the following topics:

Overview

Note: We expect the current version of CA DataMinder to integrate successfully with Autonomy ZANTAZ EAS, but this integration has not been tested."

CA DataMinder can integrate with the Autonomy ZANTAZ EAS solution. This section summarizes how emails are extracted from the EAS archive and imported into CA DataMinder. The diagram below summarizes the key components and processes involved.

For simplicity, the diagram shows a single email archive server, feeding data into a single EVF file cache. In practice, a large organization may have many such servers feeding data into multiple caches.



CA DataMinder integration with Autonomy ZANTAZ EAS

1 This server hosts the email archive solution, EAS (**1a**). This connects to an email server such as Microsoft Exchange (**2**) and archives messages in the email store (**9**).

The EAS archive solution uses an indexer process, EAS IndexerService.exe (**1b**), which in turn passes data to the External Agent API (**1c**).

The External Agent API extracts archived emails and saves them as EVF files in a cache (**3**). This cache provides the source data for the CA DataMinder Event Import utility (**4**). This utility requires the CA DataMinder infrastructure. For very large email archives, you may need to run multiple Event Import utilities simultaneously to avoid import bottlenecks.

Each Event Import utility imports archived emails into the CMS (**5**). The actual message data is not saved on the CMS; instead, a record in the CMS database for each imported email references the associated entry in the email store (**9**).

When displaying captured emails in the iConsole (**6**), the Remote Data Manager utility (**7**) retrieves data for emails archived in the email store (**9**). In the case of EAS, these data requests are sent via Microsoft IIS (Internet Information Services) (**8**).

Integration Requirements

Note the following requirements for integration with ZANTAZ EAS.

Autonomy ZANTAZ EAS

The supported versions of ZANTAZ EAS are:

- Version 4.x
- Version 5 SP1

Integration Procedure for ZANTAZ EAS

Integrating CA DataMinder with ZANTAZ EAS requires the following tasks:

1. Install the External Agent API on the EAS server.

   The External Agent API enables CA DataMinder to integrate with third party email archives such as EAS. It converts archived emails into CA DataMinder event files and saves them to a pickup folder (a shared network folder) where they can be accessed by Event Import.

   Note: Pay particular attention to step 3 of Install the External Agent API. You will need edit the EAS.INI file and run the EAS indexer process.

   You can test whether CA DataMinder console users are able to retrieve emails archived in EAS.

2. Install Event Import (see page 15).

   Event Import retrieves the extracted emails from the pickup folder specified in step 1 and imports them into the CMS. Full instructions for installing Event Import machines are given in the Event Import chapter.

3. Install the Remote Data Manager (see page 223).

   The Remote Data Manager (RDM) enables CA DataMinder to retrieve events that are archived in third party remote storage locations and display them in the iConsole or Data Management console.

   To install the RDM, you run the CA DataMinder server installation wizard. In the Remote Data Manager Configuration screen, select ZANTAZ EAS from the archive list and specify the EAS server and port number.

More information:

Event Import (see page 15)
Use the RDM to Retrieve EAS Archived Events (see page 151)

Configure EAS Integration

After installing the External Agent API, you may need to edit certain values in the External Agent API registry key. See 'Install the External Agent API' for details.

After installing your Event Import machines, you need to configure an import job.

More information:

Event Import (see page 15)

Use the RDM to Retrieve EAS Archived Events

When the External Agent API receives data from the EAS indexer process, it converts the archived email into a CA DataMinder event file along with a unique event identifier. This identifier is added to the event metadata and replicated to the CMS.

The Remote Data Manager (RDM) then uses this identifier to retrieve the e-mail from the EAS archive during subsequent event searches.

More information:

Remote Data Manager (see page 223)

Chapter 8: Symantec Enterprise Vault Integration

This section contains the following topics:

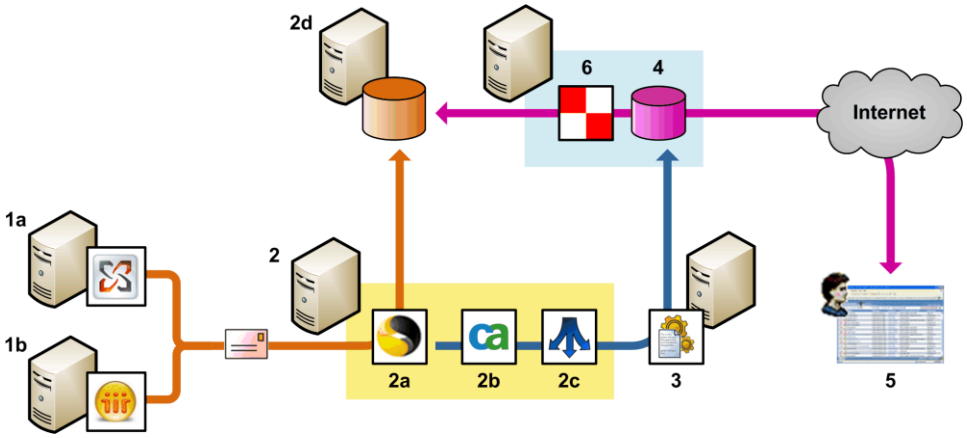
Overview

CA DataMinder can integrate with the Symantec Enterprise Vault archive solution when it is being used to archive Exchange or Domino emails.

Integration is provided through a CA DataMinder custom filter for Enterprise Vault, wgnsev.dll. In this section, the term 'EV archive agent' refers to this custom filter.

When integration is enabled, Enterprise Vault notifies the EV archive agent when it extracts an email from Microsoft Exchange or Lotus Domino. The EV archive agent passes a copy of the email to the policy engine hub.

The hub allocates the email to a policy engine, which then applies the appropriate smart tags to the email (typically an email category and a retention date) and passes this data back to the EV archive agent and then to Enterprise Vault. Finally, Enterprise Vault archives the email along with its smart tag details.



CA DataMinder integration with Enterprise Vault

Enterprise Vault extracts an email from the journal mailbox in Microsoft Exchange (**1a**) or a Lotus Domino journal (**1b**).

On this host server (**2**), Enterprise Vault (**2a**) notifies the EV archive agent (**2b**) that an email is available for processing. The EV archive agent passes the email to the hub (**2c**).

The hub allocates the email to a policy engine for processing (**3**).

The policy engine applies policy and adds smart tags to the email. These smart tags are saved as event metadata. Processed events are then replicated from the policy engine up to the CMS (**4**).

Finally, when displaying archived emails in a console (**5**), the Remote Data Manager (**6**) retrieves data for emails stored in the Enterprise Vault archive (**2d**).

About Smart Tagging

Smart Tagging is an innovative feature that enables CA DataMinder to accurately categorize events at the time of processing. The Smart Tags setting in each policy trigger defines the tag associated with that trigger. For example, you can assign to any trigger a tag such as Document Type with values of Privileged Content and Employment Solicitation. When the trigger activates, this tag is saved with the event metadata in the CMS database.

Using Smart Tagging with Enterprise Vault

Preparing your smart tags for use within Enterprise Vault, involves the following main tasks:

1. **Assign smart tag values:** In CA DataMinder, you can assign multiple values to a smart tag. If only one smart tag value is required, you can set up a smart tag rule. To do this:

   a. On the EV archive agent, create a subkey for the smart tag you want to configure specifically. The name of this subkey must match the smart tag name.

      Note: If you do not create a bespoke registry subkey for a smart tag, then the default registry settings will apply.

   b. Configure the ResolveMultipleValues registry value.

2. **Set up a 'retention smart tag':** CA DataMinder and Enterprise Vault can be used together to produce a smart tag with a retention date. This retention date specifies when Enterprise Vault will purge the archived email event.

   a. **Create Enterprise Vault retention categories:** In Enterprise Vault, configure the retention categories and their retention periods as required. By default, these categories are Business, Personal, Public and Uncategorized.

   b. **Set up a retention smart tag:** On the machine hosting the Enterprise Vault server agent, configure the RetentionSmartTag registry value.

      Note: This value must match the corresponding retention category in Enterprise Vault, as created in step 2.a.

   c. **Use retention categories to delete:** On the machine hosting the Enterprise Vault server agent, configure the RetentionSmartTag registry value to match one of the following:

      `RetentionSmartTagDiscardValue`

      `RetentionSmartTagForceDiscardValue`

Smart tags are stored by default in an Enterprise Vault property set called apmsmarttag. You can access them using their name in the following format:

```
apmsmarttag.<smarttag name>
```

You can override the name of the default propertyset by changing the value of the PropertySet registry key.

Integration Requirements

Note the following requirements for integration with Enterprise Vault.

Symantec Enterprise Vault

CA DataMinder supports Enterprise Vault 9.0 SP1 or higher for Exchange and Domino emails.

The EV archive agent (wgnsev.dll or wgnsev10.dll) must be installed on the Enterprise Vault host server.

Important! The CA DataMinder installer detects which version of Enterprise Vault is present and installs the appropriate .dll. However, if you later upgrade to Enterprise Vault 10 (see page 157) from an earlier version, you must also upgrade your CA DataMinder integration to use the correct version of the EV archive agent.

Operating System

The EV archive agent is included in the integration.msi and integration_x64.msi installation packages.

Integration.msi supports a 32-bit version of:

- Windows Server 2003 (see note 1)
- Windows Server 2008

Integration_x64.msi supports 64-bit versions of these operating systems:

- Windows Server 2003 (see note 1)
- Windows Server 2008 (see note 1)
- Windows Server 2008 R2
- Windows Server 2012

Note 1: We have not tested these operating systems with the current versions of integration.msi and integration_x64.msi.

Microsoft Exchange

The EV archive agent can process emails extracted from journal mailboxes in Microsoft Exchange Server 2003, 2007 or 2010.

Lotus Domino

The EV archive agent can process emails extracted from Lotus Domino 8.0 or 8.5 journals.

Logon account for policy engine hub service

When you install the EV archive agent, you must provide the policy engine hub service with the PE domain user credentials.

Remote Data Manager (RDM)

CA DataMinder uses the Remote Data Manager (RDM) to retrieve archived events. If using Enterprise Vault to archive:

- Exchange emails, you must install the RDM either on the Enterprise Vault host server itself or on an EV Administration Console.
- Domino emails, you must install the RDM on the Enterprise Vault host server itself.

In both cases, when you run the installation wizard, select Enterprise Vault from the archive list.

More information:

Upgrading to Enterprise Vault 10 (see page 157)

Upgrading to Enterprise Vault 10

CA DataMinder integration with Enterprise Vault is provided through two alternative versions of the EV archive agent:

- Wgnsev.dll is the default custom filter. It enables CA DataMinder to integrate with Enterprise Vault 7, 8 and 9.
- Wgnsev10.dll enables CA DataMinder to integrate with Enterprise Vault 10.

The CA DataMinder installer detects which version of Enterprise Vault is present and installs the appropriate .dll. However, if you later upgrade to Enterprise Vault 10 from an earlier version, you must also upgrade your CA DataMinder integration to use the correct version of the EV archive agent.

First, you must fix the CA DataMinder data *ingestion* capabilities for archived Domino emails. This fix enables CA DataMinder to apply policy to Domino emails before they are stored in an Enterprise Vault 10 archive. (A similar fix is not needed for archived Exchange emails.) Then you must 'repair' the CA DataMinder data *retrieval* capabilities, to enable the Remote Data Manager (RDM) to retrieve emails from an Enterprise Vault 10 archive.

To upgrade your CA DataMinder integration to support Enterprise Vault 10

1. (Needed for archived Domino emails only.) Edit the registry on the Enterprise Vault host server to specify the new .dll.

   a. Go to the following registry key:
   `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\KVS\Enterprise Vault\External Filtering\Lotus Journalling`

   b. Set the '1' registry value to:
   `<path>\wgnsev10.dll!wgnsev10.WgnSevLDfilter`

   Important! These values are case-sensitive. Type them exactly as they are shown here!

   Where <path> specifies the path to wgnsev10.dll. For example:
   `C:\Program Files (x86)\CA\CA DataMinder\client\WgnSEV10.dll!WgnSEVLD.WgnSevLDfilter`

2. Repair the CA DataMinder installation on the server that is hosting the Remote Data Manager.

   For example, in Windows Server 2008 R2, open the Programs and Features applet and select the 'CA DataMinder Server' feature. Then click the Repair button.

Integration Procedure

Integrating CA DataMinder with Symantec Enterprise Vault requires the following tasks:

1. Install the EV archive agent and a policy engine hub. You also need to deploy one or more policy engines and the RDM.
2. Manually register the EV archive agent with Enterprise Vault.
3. Configure the EV archive agent and hub and your policy engines.

   You configure the EV archive agent and policy engine hub by editing the registry on the host server. In particular, you need to:

   - Specify how the EV archive agent handles event failures and out-of-memory failures.
   - Turn on integration with CA DataMinder.
4. Install and configure the Remote Data Manager (RDM).

   CA DataMinder uses the RDM to retrieve archived events. Installing the RDM enables you to use the iConsole or Data Management console to search for emails archived in Enterprise Vault. For details, see Install the RDM.
5. Restart all Exchange or Domino journaling tasks on the Enterprise Vault server to finally to turn on integration.

More information:

Install the EV Archive Agent (see page 160)
Deploy Policy Engines (see page 160)
Register the EV Archive Agent (see page 161)

Install the EV Archive Agent

You install the EV archive agent and hub using the CA DataMinder Integration Agents installation wizard.

To install the EV archive agent

1. Log on to your Enterprise Vault host server using the Enterprise Vault service account.

   You can find this user name in the Directory Properties of the Enterprise Vault Administration Console.

2. 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.

3. Click Advanced Installation.

4. In the Advanced Install Options screen, choose Server Agents and click Install.

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

5. In the Integration Agent installation wizard, navigate to the Customer Setup screen.

6. In the Custom Setup screen, expand the Archive Agents feature and choose Symantec Enterprise Vault.

   A policy engine hub is installed automatically with this feature.

7. In the Policy Engine Hub Configuration screen, provide the credentials for the PE domain user.

   Note: The PE domain user is a domain account used by your policy engines and policy engine hub. The hub uses this account to access remote policy engine machines. For details, see the Policy Engines chapter of the *Platform Deployment Guide*.

8. In the final wizard screen, click Install to start the file transfer.

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.

Register the EV Archive Agent

The EV archive agent is a custom filter for Exchange or Domino journaling tasks that works with Enterprise Vault. After installing the EV archive agent, you must manually register this filter with Enterprise Vault. To do this, edit the registry on the agent host machine.

To register the EV archive agent

1. On the Enterprise Vault server, you need to modify certain registry values.

   The required registry key depends on whether the host server is running a 32-bit or 64-bit operating system. Details about the supported operating systems are in Integration Requirements (see page 156).

   32-bit versions of Windows Server

   If archiving Exchange emails, locate or create this registry key:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\KVS\Enterprise Vault
     \External Filtering\Journaling
   ```

   If archiving Domino emails, locate or create this registry key:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\KVS\Enterprise Vault
     \External Filtering\Lotus Journaling
   ```

   64-bit versions of Windows Server

   If archiving Exchange emails, locate or create this registry key:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\KVS\Enterprise Vault
     \External Filtering\Journaling
   ```

   If archiving Domino emails, locate or create this registry key:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\KVS\Enterprise Vault
     \External Filtering\Lotus Journaling
   ```

   In all cases, if the \Journaling or \Lotus Journaling registry key does not already exist, you must create it.

2. Within this registry key, add a new string value with these details:

   Name: Set this to be an integer such as 1, 2 or 3. The number determines the filter processing order, with registry value 1 processed first. But see the warning below.

   Type: REG_SZ

   Data: The required data depends on whether Enterprise Vault is archiving Exchange or Domino emails. If archiving:

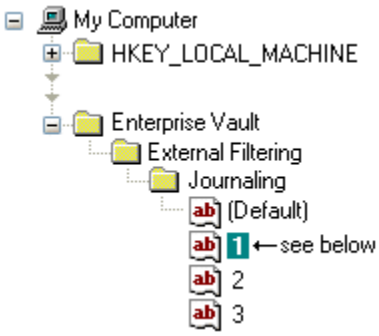
   - **Exchange emails**, set this registry value to:
     `WgnEVFilter.WgnEnterpriseVaultFilter`
   - **Domino emails**, set this registry value to:
     `<path>\WgnSEV.dll!WgnSEVLD.WgnSevLDfilter`

     Where <path> is the full path to WgnSEV.dll, for example:
     `c:\program files\CA\CA DataMinder\client\WgnSEV.dll!WgnSEVLD.WgnSevLDfilter`

   In both cases, keep the uppercase and lowercase letters exactly as shown for the filter elements (the path element for Domino emails is case-insensitive).

   Important! It is essential that the EV archive agent is processed before the Compliance Accelerator Journaling Connector. This means that when defining numeric registry values, the EV archive agent must have a lower number than the Journaling Connector.

   For example, if the Journaling Connector is already assigned to registry value '1' you must rename this value to '2' and create a new value '1' for the EV archive agent. To identify which numeric registry value is associated with the Journaling Connector, look for a value whose data is set to KVS.Accelerator.Plugin.Filter.



   Example Enterprise Vault registry values

   In this example, the filter for the EV archive agent is assigned to registry value '1'. Enterprise Vault is archiving Exchange emails so the registry value data is set to:
   `WgnEVFilter.WgnEnterpriseVaultFilter`

   If Exchange or Domino journaling tasks use multiple filters, you must decide in which order the EV archive agent is processed—but see the warning above.

3. Restart all Exchange or Domino journaling tasks on the Enterprise Vault server.

   This completes the registration of the EV archive agent. Now you must configure the EV archive agent, the hub, and the policy engines. See the following sections for details.

More information:

Integration Requirements (see page 156)
Turn On Enterprise Vault Integration (see page 175)

Configure Enterprise Vault Integration

You configure the EV archive agent and policy engine hub by editing the registry on the host server. In particular, you need to specify how the EV archive agent handles event failures and out-of-memory failures.

Configure the Policy Engine Hub

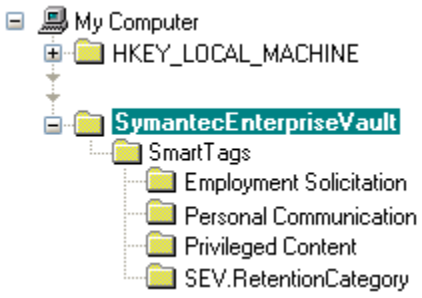
This is described in Configure the policy engine hub (see page 302).

Configure the EV Archive Agent

To configure CA DataMinder integration with Enterprise Vault, you need to modify registry values in the following registry key:

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\CA DataMinder
  \CurrentVersion\ExternalIntegration\SymantecEnterpriseVault
```

The key structure is shown below:



SymantecEnterpriseVault Key

The SymantecEnterpriseVault registry key contains the following optional registry values plus the SmartTags subkey.

OperationMode

Type: REG_SZ

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

EnableIntegration

Type: REG_DWORD

Data: Defaults to 0. To enable integration, set this registry value to 1. This ensures that the EV archive agent processes all emails received from Enterprise Vault. Set this registry value to 0 to disable integration with Enterprise Vault.

HubFailureMode

Type: REG_SZ

Data: Defaults to Fail. This specifies what happens if the HighWaterMarkEventCount or HighWaterMarkMB thresholds are exceeded. If set to:

- **Wait:** The hub stops accepting emails from the EV archive agent until the event queue shortens, or the allocated memory amount falls 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 exceeded. Email failures are returned to the EV archive agent. Normal operations resume when the event queue shortens, or the allocated memory amount falls 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 EV archive agent to only log errors or important system messages. Log entries are written to the wgnsev_<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—see below. The supported logging levels are:

1 Errors only

2 Errors and warnings

3 Errors and warnings, plus informational and status messages

LogFilePath

Type: REG_SZ

Data: Defaults to empty. This specifies the folder you want to write log files to. If the path is not defined, the log file is saved to the \System\Data\Log subfolder in the CA DataMinder installation folder.

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 EV archive agent creates a new log file.

Log entries are written to a wgnsev_<date>.log file—for details see LogLevel above.

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 latest file reaches its maximum permitted size (see LogMaxSizeBytes), the oldest log file is deleted to enable a new one to be created.

DiagnosticFolder

Type: REG_SZ

Data: Specifies the path and folder where diagnostic files will be saved. The creation of these files is determined by the CreateEVF registry value described below.

Note: This folder is not created automatically.

CreateEVF

Type: REG_DWORD

Data: Specifies whether an EVF file (containing event and archive identifiers) is created for each email processed. If this value is set to:

0 The EV archive agent never creates EVF files.

1 The EV archive agent always creates EVF files.

2 It only creates EVF files if an error occurs when extracting the contents of an email.

Note: Any EVF files created are saved in the DiagnosticFolder—see above.

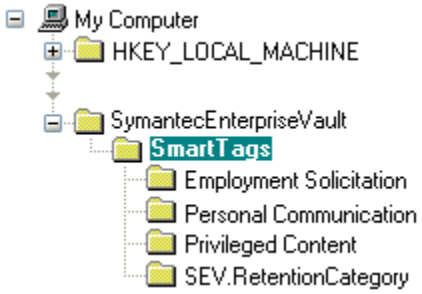
PropertySet

Type: REG_SZ

Data: Specifies the APM smarttag property set to be used. The default value is apmsmarttag.

SmartTags key

The SmartTags subkey contains the following optional registry values, plus a <Smart tag> subkey for each smart tag specified in user policy. Registry values in these keys determine how a smart tag is stored in the Enterprise Vault archive and how CA DataMinder resolves multiple values for a single smart tag type.



The following registry settings are available:

ConcatenateSeparatorDefault

Type: REG_SZ

Data: Defaults to a comma. Specifies the separator character between multiple smart tag values if ResolveMultipleValuesDefault is set to Concatenate.

SmartTagIgnoreList

Type: REG_SZ

Data: Specifies a list of smart tags that are ignored when archiving the email into Enterprise Vault. In effect, this registry value allows you to disregard specific smart tags without needing to add an associated smart tag subkey with the registry value SuppressAttribute set to 1.

IgnoreUnlistedSmartTags

Type: REG_DWORD

Data: Defaults to 0. If you set this to zero, all smart tags are archived with the event in Enterprise Vault, even if a smart tag has no subkey defined. If set to 1, only smart tags with their own designated subkey are archived in Enterprise Vault.

FailureCodeSmartTag

Type: REG_SZ

Data: Defaults to WgnFailureCode. This value provides the name of the Enterprise Vault attribute applied to events that are failed by CA DataMinder, but go on to be processed by Enterprise Vault. The attribute value contains the error code and string.

ResolveMultipleValuesDefault

Type: REG_SZ

Data: Specifies how policy engines handle multiple smart tag values. That is, if an email activates multiple triggers that all apply the same smart tag name but different smart tag values, ResolveMultipleValuesDefault determines which values are archived with the email.

This registry value provides the default handling for all smart tags. If the ResolveMultipleValuesDefault registry value is not specified in a <Smart tag> subkey, ResolveMultipleValuesDefault determines how multiple tag values are handled.

Available options include:

Raw

This is the default option. It allows the filter to pass smart tags with multiple values directly.

Concatenate

Multiple smart tag values are concatenated, using the specified ConcatenateSeparator.

Priority

The policy engine only saves the smart tag value listed first in PriorityOrder. Other values are ignored. All values are ignored if the PriorityOrder registry value is not defined.

Note: Two other options are supported: 'Highest' and 'Lowest'. These can only be used if a smart tag only returns numeric values. These options save only the highest or lowest value, respectively.

RetentionSmartTag

Type: REG_SZ

Data: Specifies the smart tag name used by policy engines to save an event's minimum retention period. For example, set this registry value to sev_retention_category. You must also ensure that this name is added to the Smart Tags setting for all triggers in the user policy that are configured to apply retention smart tags.

To configure Enterprise Vault to delete emails rather than archive them, you need to set this to the same value as RetentionSmartTagDiscardValue or RetentionSmartTagForceDiscardValue.

Note: Note the following:

- If multiple retention smart tag values are returned, the filter automatically selects the value with the longest retention period.
- The retention date specifies when the event becomes eligible for purging. If you do not configure a 'retention period' smart tag for use by CA DataMinder policy engines, Enterprise Vault applies a default retention period to all emails in the archive.

RetentionSmartTagDiscardValue

Type: REG_SZ

Data: This is set to Discard when CA DataMinder integration with Enterprise Vault is installed. It specifies that Enterprise Vault will not archive, but delete any email with only this retention tag. But if the email triggers multiple rules and has:

- Either, two or more retention tags (that is, a 'Discard' tag plus one or more additional tags), then Enterprise Vault ignores 'Discard' and *does* archive the email, assigning the retention category with the longest retention date.
- Or, a 'Discard' tag plus an unrecognized retention tag (for example, because no matching retention category exists), Enterprise Vault ignores 'Discard' and does archive the email, assigning the default retention category.

RetentionSmartTagForceDiscardValue

Type: REG_SZ

Data: This is set to Force Discard when CA DataMinder integration with Enterprise Vault is installed. It specifies that instead of archiving, Enterprise Vault will delete *any* email with this retention tag. If an email has any additional retention categories assigned, these are ignored.

EmptySmartTagReplacementValue

Type: REG_SZ

Data: Enterprise Vault does not store custom attributes with no value. If you are expecting events with an empty smart tag value and want Enterprise Vault to store those events with the smart tag, use this registry value to specify a replacement value for the empty smart tag value.

Note: If you do not set this registry value, Enterprise Vault will process the events, but will not store a smart tag with an empty value.

<Smart tag> Subkeys

These subkeys allow default settings to be superseded, if required. In the example registry diagram for the SmartTags key, Enterprise Vault integration supports four smart tag subkeys:

- employment_solicitation
- personal_communication
- privileged_content
- SEV.RetentionCategory

You must create smart tag subkeys manually. Each subkey must contain the ResolveMultipleValues registry value; other registry values are optional. If they are not specified, CA DataMinder uses the corresponding default registry values in the parent SmartTags key.

The full range of registry values are described below:

SuppressAttribute

Type: REG_SZ

Data: Defaults to 0. If set to 0, the corresponding smart tag *is* stored as an Enterprise Vault custom attribute. If set to 1, the corresponding smart tag is **not** stored as an Enterprise Vault custom attribute.

PriorityOrder

Type: REG_SZ

Data: Specifies a comma separated list of smart tag values in decreasing priority. That is, if ResolveMultipleValues is set to Priority, CA DataMinder saves the first tag value in this list with the email when archiving; if the highest priority tag value is not detected (because no trigger applied this value), CA DataMinder saves the second tag value in the list with the email when archiving, and so on.

ConcatenateSeparator

Type: REG_SZ

Data: Defaults to a comma. Specifies the separator character between multiple smart tag values if ResolveMultipleValues is set to Concatenate.

ResolveMultipleValues

Type: REG_SZ

Data: Defaults to Raw. This specifies how policy engines handle multiple smart tag values. That is, if an e-mail activates multiple triggers that all apply the same smart tag name but different smart tag values, ResolveMultipleValues determines which values are archived with the e-mail. Available options include:

Raw

This is the default option. Allows the filter to pass smart tags with multiple values directly.

Concatenate

Multiple smart tag values are concatenated, using the specified ConcatenateSeparator.

Priority

The policy engine only saves the smart tag value with the highest priority, as specified by PriorityOrder. Other values are ignored. All values are ignored if the PriorityOrder registry value is not defined.

Note: Two other options are supported: 'Highest' and 'Lowest'. These can only be used if a smart tag only returns numeric values. These options save only the highest or lowest value, respectively.

More information:

SmartTags key (see page 166)

Configure the Domino Journal Task

After installing the EV archive agent, you may need to configure the Domino journal task to allow the agent to ingest emails into the Enterprise Vault archive.

To configure the Domino journal task

(This task is not necessary if you are using Enterprise Vault 9.0 SP1)

You must update your Domino journal task configuration file, EVLotusDominoJournalTask.exe.config. The simplest method is to merge an example configuration file supplied with CA DataMinder into your existing configuration file. You may also need to edit the configuration file to reflect the version of Enterprise Vault that you are using.

1. Merge the <assemblyBinding> element from the example configuration file into the <runtime> element of your existing configuration file.

   By default, your existing configuration file is in C:\Program Files\Enterprise Vault on the Enterprise Vault server. The example configuration files are available on the CA DataMinder page of the CA Support Online site. If you do not have an existing configuration file:

   a. Select the appropriate example file (there are separate files for Enterprise Vault 9.0 GA and SP2).

   b. Copy the appropriate entire example file into C:\Program Files\Enterprise Vault.

2. For each assembly in the configuration file, ensure that the *newVersion* parameter in each <bindingRedirect> element matches the version of Enterprise Vault that you are using (the file contents are shown below).

   If you are using Enterprise Vault 9.0 SP1, no file changes are needed.

   If you are using any other version of Enterprise Vault 9.0:

   a. Locate each <binding Redirect> line in the configuration file:

   b. For Enterprise Vault 9.0 GA, ensure the *newVersion* parameters are set to:
   `<bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.0.0" />`

   For Enterprise Vault 9.0 SP2, ensure the *newVersion* parameters are set to:
   `<bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.2.0" />`

   For future, as-yet-unreleased service packs (for example, Enterprise Vault 9.0 SP3), ensure the *newVersion* parameters match the service pack number. In the case of SP3, ensure the parameters are set to:
   `<bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.3.0" />`

3. Save the changes to EVLotusDominoJournalTask.exe.config

Example configuration file

The contents of the example EVLotusDominoJournalTask.exe.config configuration file are shown below:

```
<configuration>
  <runtime>
    <generatePublisherEvidence enabled="false"/>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="KVS.EnterpriseVault.LotusDominoInterfaces"
         publicKeyToken="26c5e2ccf2b9267c" />
        <bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.2.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="KVS.EnterpriseVault.Interop.IndexClient"
         publicKeyToken="26c5e2ccf2b9267c" />
        <bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.2.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity
name="KVS.EnterpriseVault.Interop.EVContentManagementAPI"
         publicKeyToken="26c5e2ccf2b9267c" />
        <bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.2.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="KVS.EnterpriseVault.Interop.RetentionAPI"
         publicKeyToken="26c5e2ccf2b9267c" />
        <bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.2.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

Configure the Remote Data Manager

CA DataMinder uses the Remote Data Manager (RDM) to retrieve archived events. When the policy engine has processed an email event and policy has caused it to be saved, the event metadata and an event identifier are replicated to the CMS. The event identifier allows the RDM to retrieve the email from the Enterprise Vault archive when a reviewer searches for archived searches using the iConsole or Data Management console.

Before CA DataMinder can use the RDM to retrieve:

- Archived Domino emails, you must securely cache the password for the default Notes user on your Enterprise Vault server.
- Archived Exchange emails, you must copy a DLL onto the EV Administration Console (also referred to as a VAC machine).

Register the Enterprise Vault DLLs

After installing the EV archive agent, you must register two of the EV runtime DLLs.

To register the Enterprise Vault DLLs

1. On the Enterprise Vault host server, launch the .NET Framework 2.0 Configuration snap-in.

   Find this snap-in in Administrative Tools.

2. In the .NET Framework 2.0 Configuration window, right-click the Assembly Cache and click Add.
3. In the Add an Assembly window, browse to the Enterprise Vault installation folder.

   The default location of this folder is:

   - **32-bit servers:** C:\Program Files\Enterprise Vault
   - **64-bit servers:** C:\Program Files (x86)\Enterprise Vault

4. Select the following DLLs:

   KVS.EnterpriseVault.Interop.EVContentManagementAPI.dll

   KVS.EnterpriseVault.Interop.IndexClient.dll

5. Click the Open button.
6. Restart the CA DataMinder infrastructure service.

Securely Store the Password for the Default Notes User

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 password for the default Notes user. The RDM uses this account to search Enterprise Vault for archived Domino emails.

You must run wgncred.exe on the Enterprise Vault host server. Wgncred.exe is installed when you install the EV archive agent. Find wgncred.exe in the \System subfolder of the CA DataMinder installation folder.

To set the password for the default Notes user

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 Symantec Enterprise Vault for Lotus Domino (RDM) component.

   The component ID is RDMSEVLDClient.
3. Type the password for this component.

To clear the password for the default Notes user

If you need to reset the stored password for the default Notes user, you must first clear the original password.

1. From a command prompt, run:
   ```
   wgncred -clear
   ```
   A list of components is displayed.
2. Choose the Symantec Enterprise Vault for Lotus Domino (RDM) component.

   The component ID is RDMSEVLDClient.

Note: For full details about wgncred.exe, see the Technical Information chapter of the *Platform Deployment Guide*.

Turn On Enterprise Vault Integration

After installing, registering, and configuring the various components, you must restart all Exchange or Domino journaling tasks on the Enterprise Vault server. This will complete the registration of the EV archive agent and turn on integration with CA DataMinder. As soon as registration is complete, Enterprise Vault notifies the EV archive agent that emails are available for processing. The EV archive agent then passes these emails to the policy engine hub.

The filter for the EV archive agent will be applied to all journaling tasks on the Enterprise Vault server.

Troubleshooting

Searches Fail to Retrieve Archived Emails

Symptom:

When a reviewer searches for Domino emails archived in EnterpriseVault, the search fails to retrieve any emails. Both the iConsole and Data Management console display a Failure to Retrieve Mail notification message. Entries are also written to the CA DataMinder activity log on the RDM host server. These log entries are:

- Failed to load assembly
- Unable to retrieve email

Note: In the current CA DataMinder release, log files are typically stored in CA's \data\log subfolder of the Windows All Users profile.

Solution:

Confirm that wgninfra.exe.config is correctly configured. In particular, confirm that the <codeBase> lines specify the correct paths to:

- KVS.EnterpriseVault.Interop.EVContentManagementAPI.dll
- KVS.EnterpriseVault.Interop.IndexClient.dll

Searches Fail After Installing SEV Service Pack

Symptom:

After installing an Enterprise Vault service pack, when a reviewer searches for Domino emails archived in EnterpriseVault, the search fails to retrieve any emails.

This problem occurs because the new versions of the Enterprise Vault assemblies no longer match the versions in wgninfra.exe.config. Consequently, the Remote Data Manager can no longer load the required assemblies.

Both the iConsole and Data Management console display a Failure to Retrieve Mail notification message. Entries are also written to the CA DataMinder activity log on the RDM host server. These log entries are:

- Failed to load assembly
- Unable to retrieve email

Note: In the current CA DataMinder release, log files are typically stored in CA's \data\log subfolder of the Windows All Users profile.

Solution:

Repair the Remote Data Manager (RDM) and then reconfigure the Domino journal task.

To repair the CA DataMinder server.msi image.

1. On the RDM host server, use the Add or Remove Programs applet to change the CA DataMinder Server program.
2. When the installation wizard starts, go to the Program Maintenance page and click Repair.

   The wizard examines the newly installed Enterprise Vault assemblies and updates the information in wgninfra.exe.config.

To reconfigure your Domino journal task configuration file

1. Find the journal task configuration file, EVLotusDominoJournalTask.exe.config.

   By default, this file is in C:\Program Files\Enterprise Vault on the Enterprise Vault server.
2. For each assembly in the configuration file, verify that the *newVersion* parameter in each <bindingRedirect> element matches the version of Enterprise Vault that you are using (the file contents are shown below). For example:
   - If you installed SP2 for Enterprise Vault 9.0, verify that the *newVersion* parameters are set to:
     ```
     <bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.2.0" />
     ```
   - For future, as-yet-unreleased service packs (for example, Enterprise Vault 9.0 SP3), verify that the *newVersion* parameters match the service pack number. In the case of SP3, verify that the *newVersion* parameters are set to:

```
        <bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.3.0" />
```

3. Save the changes to EVLotusDominoJournalTask.exe.config.

Example File: EVLotusDominoJournalTask.exe.config

Example configuration file

The contents of the example EVLotusDominoJournalTask.exe.config configuration file are shown below:

```
<configuration>
  <runtime>
    <generatePublisherEvidence enabled="false"/>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="KVS.EnterpriseVault.LotusDominoInterfaces"
         publicKeyToken="26c5e2ccf2b9267c" />
        <bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.2.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="KVS.EnterpriseVault.Interop.IndexClient"
         publicKeyToken="26c5e2ccf2b9267c" />
        <bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.2.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity
name="KVS.EnterpriseVault.Interop.EVContentManagementAPI"
         publicKeyToken="26c5e2ccf2b9267c" />
        <bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.2.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="KVS.EnterpriseVault.Interop.RetentionAPI"
         publicKeyToken="26c5e2ccf2b9267c" />
        <bindingRedirect oldVersion="9.0.1.0" newVersion="9.0.2.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

Chapter 9: EMC SourceOne Integration

This section contains the following topics:

Overview

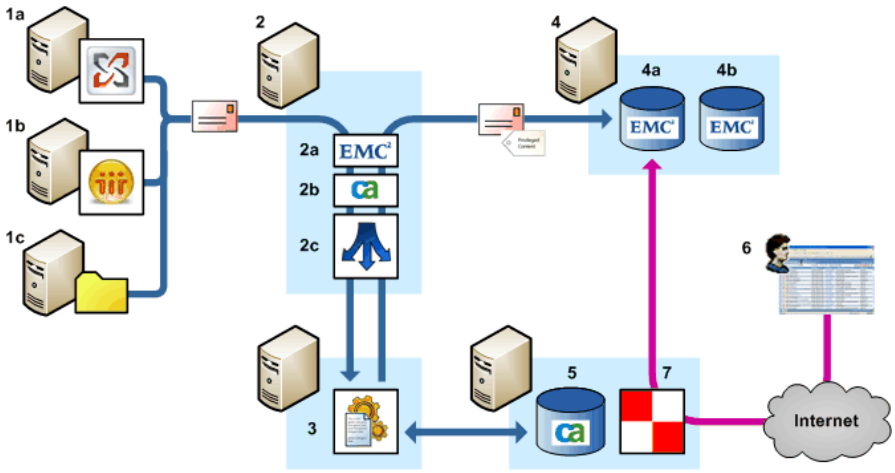
CA DataMinder can integrate with the EMC SourceOne archiving solutions for Microsoft Exchange and Lotus Notes Domino. It can also integrate with EMC SourceOne when it is being used to archive internet emails.

Integration is provided through a business component extension (BCE) for SourceOne, wgnemcs1.dll. In This section, the term 'SourceOne archive agent' refers to this BCE.

How does the integration work? SourceOne extracts emails from Exchange or Domino or from an internet email drop folder. They are then passed to CA DataMinder policy engines for analysis and processing. When processing is complete, the emails, plus any resulting smart tags, are returned to SourceOne for storage in the archive. SourceOne can then use these smart tags to distribute emails across different archives based on criteria indicated by the smart tags. For example, you can use smart tags to indicate archive retention periods.

Note: Internet emails are stored as EML files and conform to RFC 2822. They are commonly referred to as internet emails because of the RFC document, 'Internet Message Format', that describes them.

The key components are illustrated in the diagram below. For simplicity, this diagram shows SourceOne installed on a single server, and the SourceOne archive agent passing emails to a single CA DataMinder policy engine. In practice, large organizations may have a distributed SourceOne deployment and use multiple policy engines.



CA DataMinder integration with SourceOne archive

1. **Email source.** SourceOne extracts emails from an Exchange journal mailbox (**1a**) or, for Domino, from the EMC Mail Journal Database (**1b**). Or it extracts internet emails from a drop folder (**1c**).
2. **EMC SourceOne management server.** SourceOne (**2a**) extracts emails from the journal mailbox or journal database and passes them to the SourceOne archive agent (**2b**). In turn, the archive agent passes emails to a policy engine hub (**2c**). The hub then distributes emails to policy engines (**3**).
3. **Policy engine**. The policy engines analyzes emails, applies policy triggers, and attaches smart tags as necessary.
4. **EMC SourceOne archive.** Processed emails plus any resulting smart tags are returned to SourceOne for archiving. This example shows two archives, one for storing emails with short retention periods (**4a**) and the other used for storing emails with longer retention periods (**4b**).
5. **CMS.** Events generated as a result of policy processing, including metadata that incorporates the unique message identifiers, are replicated to the CMS and stored in the database. In this example, the CMS machine also hosts the Remote Data Manager (**7**).
6. **iConsole.** Reviewers can search for archived emails.
7. **Remote Data Manager.** When a user searches for archived emails in the iConsole (**6**), the Remote Data Manager retrieves data for these emails from the archive (**4**).

Integration Requirements

Note the following requirements for integration with EMC SourceOne.

EMC SourceOne Email Management for Exchange or Domino

CA DataMinder supports SourceOne 6.6 SP1 or later. But see the note.

Note: CA DataMinder does not support SourceOne 6.8.2 due to incompatible product changes introduced in this version.

Integration Procedure for SourceOne

Integrating CA DataMinder with SourceOne involves the following tasks:

1. (SourceOne for Domino only) Ensure that the EMC SourceOne Extension for Lotus Domino is installed on your Domino server.
2. Install the SourceOne archive agent on your SourceOne Worker server. If your SourceOne solution is deployed across multiple Worker servers, you must deploy the SourceOne archive agent on each of those servers.
3. Configure the SourceOne Archive Agent.
   - (Lotus Notes email only) Set the NotesTemplateFile registry.
   - (Microsoft Exchange email only) No additional configuration to the SourceOne archive agent is necessary, though you can customize the logging level if required.
4. Deploy your CA DataMinder policy engines.
5. Configure the CA DataMinder Business Component for SourceOne.
6. Add CA DataMinder smart tags to SourceOne filtering rules.
7. Deploy the Remote Data Manager to allow users to search for archived emails and review them in the iConsole or Data Management console.

More information:

Install the SourceOne Extensions for Domino (see page 182)
Install the SourceOne Archive Agent (see page 182)
Configure the SourceOne Archive Agent (see page 183)
Deploy Policy Engines (see page 185)
Configure SourceOne Business Components (see page 186)
Add CA DataMinder Smart Tags to SourceOne Filtering Rules (see page 187)
Use the RDM to Retrieve SourceOne Archived Events (see page 188)

Install the SourceOne Extensions for Domino

To enable SourceOne for Domino to extract emails from Domino, the 'EMC SourceOne Extension for Lotus Domino' must be installed on your Domino server. This extension allows you to specify the EMC Mail Journal Database as the source for the Domino emails that you want to extract into your SourceOne archive

Ensure that this extension is installed before you proceed with the CA DataMinder-SourceOne for integration.

Install the SourceOne Archive Agent

You need to install the SourceOne archive agent on your SourceOne Worker server. If your SourceOne solution is deployed across multiple Worker servers, you must deploy the SourceOne archive agent on each of those servers. A policy engine hub is installed automatically with the SourceOne archive agent.

To install the SourceOne Archive 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 Archive Agents feature and EMC SourceOne.

   A policy engine hub is installed automatically with this agent.

6. In the Policy Engine Hub Configuration screen, provide the credentials for the PE domain user.

   In addition to the usual requirements for the PE domain user, this user account must also be able to access the SourceOne server. In fact, we recommend that you use the SourceOne service account as your PE domain user.

7. In the final wizard screen, click Install to start the file transfer.

Configure the SourceOne Archive Agent

If EMC SourceOne is used to archive:

- Microsoft Exchange emails, no further configuration is necessary after installing the SourceOne archive agent.
- Lotus Notes emails, verify that the NoteTemplateFile registry value is set correctly.

When processing Lotus Notes emails, the SourceOne archive agent must copy the items into a temporary notes database. The file specified in the NotesTemplateFile registry is used as the template when creating this temporary database. Using a template minimizes the risk of mail items being failed by the SourceOne Archive Agent when the mail is missing mail properties.

If you later need to designate a different SourceOne server or you want to change the logging level or log file location, you can modify registry values on the RDM server and SourceOne management server.

Registry values for the SourceOne archive agent are in this registry key:

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\CA DataMinder
  \CurrentVersion\EMCSourceOneAdapter
```

Within this key, edit the following registry values:

DeDuplicationMode

Type: REG_DWORD

Machine: Edit this registry value on the machine where the EMC SourceOne journaling task is running.

Data: Deduplication lets archive integration skip emails that are defined as duplicates by EMC SourceOne. Policy is not applied to duplicates and they are not captured for review. Deduplication is disabled by default (zero).

0

Disables deduplication.

1

Enables deduplication for an email if it already exists in any of the SourceOne archive folders into which it has been archived.

DisableProcessing

Type: REG_DWORD

Machine: Edit this registry value on the machine where the EMC SourceOne journaling task is running.

Data: By default, the adapter is enabled (zero). Set this to a non-zero value to disable the adapter. If disabled, the adapter creates log entries for non-processed messages and does not pass any messages to the policy engine. This value is tested for each mail and therefore can be switched on and off at will.

DMSServer

Type: REG_SZ

Machine: Edit this registry value on the RDM server only.

Data: Set this registry value to the name or IP address of the SourceOne machine which hosts the Document Management Service.

The RDM needs to be able to identify the SourceOne Document Management Service host. If you change the SourceOne Document Management Service host after you install the RDM, edit the DMSServer registry value to specify the new host machine.

LogLevel

Type: REG_DWORD

Machine: Edit this registry value on the RDM server and on the server hosting the SourceOne archive agent (this is the SourceOne management server).

Data: Defaults to 2. This registry value determines the level of logging for message processing. For example, you can configure the archive agent to only log errors or important system messages.

Log entries are written to wgnemcs1_<date>.log where <date> is the date and time when the log file was created.

The supported logging levels are:

1

Errors only.

2

Errors and warnings.

3

Errors, warnings, plus informational and status messages.

MaxNotesInDbase

Type: REG_DWORD

Machine: Edit this registry value on the machine where the EMC SourceOne journaling task is running.

Data: Limits the number of emails in the temporary .NSF file that CA DataMinder creates for message processing. If the value is exceeded, a new .NSF file is created. The old file is deleted after the Policy Engine has finished processing any mails in it. The default value is 1000 mails; typically, this default suffices to keep the temporary file's size around 100MB.

NotesTemplateFile

Type: REG_SZ

Machine: Edit this registry value on all RDM and SourceOne Worker Servers.

Data: Set this registry value to a fully qualified folder path to one of the following Lotus Notes Template files:

- (for Domino 8) mail8.ntf
- (for Domino 8.5) mail85.ntf

When you use EMC SourceOne to archive Lotus Notes emails, we recommend you set this registry value to prevent occasional errors with Notes that do not have certain properties specified.

LogFilePath

Type: REG_SZ

Machine: Edit this registry value on the RDM server and on SourceOne management server.

Data: This registry value redirects the wgnemcs1 log file to an alternative location (see the LogLevel description for the default location). Set this registry value to a fully qualified folder path to an existing.

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.

Configure SourceOne Business Components

Configure SourceOne to allow integration with CA DataMinder and to use CA DataMinder smart tags to distribute emails across different archives.

To integrate SourceOne with CA DataMinder

Configure the SourceOne activity for an organizational policy to use the CA DataMinder business component:

1. In the EMC SourceOne console, expand the Organizational Policies branch and locate the required policy.
2. In the Edit Activity wizard:
   a. Go to the Business Components page.
   b. Select the CA DataMinder SourceOne extension.
   c. Specify the evaluation order so that the CA DataMinder SourceOne extension is listed above the Address Rules extension.

Add CA DataMinder Smart Tags to SourceOne Filtering Rules

You now need to use CA DataMinder smart tags to distribute emails across different archives.

You can edit the filtering rules for a SourceOne activity to use smart tags attached to emails by CA DataMinder.

To add CA DataMinder smart tags to SourceOne filtering rules

In the example below, CA DataMinder attaches a smart tag named 'Retention' to the emails that it processes. This smart tag has a value of 'Retain this item'.

1. Set up your CA DataMinder user policies to apply smart tags to events captured by CA DataMinder.

   For details, refer to the 'Smart Tagging' chapter in the CA DataMinder *Policy Guide*.

2. In the Edit Activity wizard, go to the Filtering Rules page.
3. Edit the filtering rule that you want to use.
4. Define the rule criteria. Specifically, edit the following item:

   'Messages with <custom metadata>'

5. In the Custom Metadata dialog, specify the following items:

   Field Name

   Enter the *name* of the CA DataMinder smart tag. For example, enter 'Retention'.

   Condition

   Select the required condition for the metadata value. For example, choose 'Equals' or 'Contains'.

   Value

   Enter the *value* of the CA DataMinder smart tag. For example, enter 'Retain this item'.

   Note: Do not put quotes around the Field Name or Value, even if the name or value contains spaces.

Use the RDM to Retrieve SourceOne Archived Events

When SourceOne passes emails to the SourceOne archive agent, each email includes a unique identifier. In turn, the archive agent passes this identifier to a policy engine. When the policy engine processes the email from SourceOne, it generates a CA DataMinder email event and adds the unique identifier to the event's metadata. The event and its identifier are then replicated to the CMS. The RDM will use this identifier to retrieve the email from the SourceOne archive during subsequent event searches.

You can install the RDM on any CA DataMinder server. The architecture diagram in the Overview (see page 179) shows the RDM installed on the CMS server.

To set up event retrieval

1. Install the RDM (see page 225).
2. (Applies only if the RDM is running remotely and *not* installed on your SourceOne management server)

   Install the EMC SourceOne SRE Initialization package (ES1_SRESetup.exe). This package enables the RDM to connect to SourceOne.

   Obtain the package from EMC. The minimum supported SRE version is 6.51.1317.

   Note: Microsoft .NET Framework 3.0 is a pre-requisite for the SRE.

More information:

EMC SourceOne Integration (see page 231)

Chapter 10: External Agent API

The External Agent API facilitates third party integration with CA DataMinder and smart tag metadata. It enables third party applications to pass messages to CA DataMinder for policy processing and, if required, to store the resulting smart tag metadata with the archived message. Events corresponding to these policy-processed messages can also be stored on a CMS. It can also convert archived emails to CA DataMinder event (EVF) files and save them to a shared network folder where they can be accessed by the Event Import and imported onto a CMS.

Note: When generating EVF files, the External Agent API currently supports Zantaz Exchange Archive Solution (EAS). These instructions assume you are familiar with the configuration and operation of EAS.

A 3rd-party integrator can also use the External Agent API for custom archives (see page 196).

This section contains the following topics:

Output Destinations

The External Agent API can directly generate EVF files and output them to a file cache (that is, a folder). Or it can send emails for policy processing directly to a local policy engine or to a local policy engine hub. The output destination is specified using interfaces obtained when you CoCreate the relevant COM object. The External Agent API includes two sets of interfaces, available by different COM objects:

- **WgnActiveImportConnector:** Use these interfaces to apply policy to messages passed to the External Agent API by a third party application.
- **WgnImportConnector:** Use these interfaces to generate EVFs from messages passed to the External Agent API by a third party archive.

Full details for configuring the External Agent are in the *External Agent COM API Specification*—see the next section.

Integrating Programmatically with the External Agent API

The interfaces, methods and parameters used by the External Agent API to integrate with third party applications are described in the *External Agent COM API Reference Guide.*

External Agent API Requirements

Note the following requirements for the External Agent API:

Operating system

The External Agent API is included in the integration.msi and integration_x64.msi installation packages.

Integration.msi supports a 32-bit version of:

- Windows Server 2003 (see note 1)
- Windows Server 2008

Integration_x64.msi supports 64-bit versions of these operating systems:

- Windows Server 2003 (see note 1)
- Windows Server 2008 (see note 1)
- Windows Server 2008 R2
- Windows Server 2012
- **Note 1:** We have not tested these operating systems with the current versions of integration.msi and integration_x64.msi.

Email solution

The External Agent host machine must have either an Exchange compatible application or Lotus Notes installed.

Microsoft Exchange

For example, Outlook 2003. Furthermore, this must be the default email application on the target machine.

Lotus Notes

Versions 6, 6.5, 7 or 8.

File system

(Applicable to the WgnImportConnector interfaces only)

For NTFS file systems, we recommend that you disable creation of 8.3 file names on the External Agent API host machine. If 8.3 file creation remains enabled, performance may be adversely affected.

Compatibility with Event Import

The version of the External Agent API must match that of Event Import.

For example, if Event Import 6.0 is installed, you must use version 6.0 of the External Agent API. Also, Event Import must be configured to import EVFs.

EAS Integration

If you use the External Agent API to convert emails from an EAS archive into EVF files, the API requires the EAS indexer process IndexerService.exe, versions 3.2.1.992, 4.2.0.1624, or 4.3.1.1962.

Uninstalling

If the External Agent API and CA DataMinder (whether a server or client machine) are installed on the same machine but you subsequently uninstall the External Agent API, you must repair the remaining CA DataMinder installation to ensure it operates correctly after removing the External Agent API.

Install the External Agent API

You install the External Agent API with the CA DataMinder Integration Agents installation wizard.

To install the External Agent API

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
      ```
10. (Required for EAS integration only) If you are integrating with an EAS archive, follow these steps:
    a. Find EAS.ini in the Windows ('systemroot') folder of the EAS machine.
    b. Edit EAS.ini and add the following line to the [FULLTEXT] section:

       ```
       ComplianceDLL=<path>\wgnrdi.dll
       ```

       Where <path> is the full path to Wgnrdi.dll (that is, the External Agent API), installed in step 5.
    c. Run the EAS indexer process.

       The External Agent API to start converting messages from the email archive and saving them in the EVF file cache.
11. The External Agent API installation is now complete.

EVF File Cache Guidelines

By default, the installation batch file sets the EVF file cache to be the \client\RDIcache subfolder of the CA DataMinder installation folder on the External Agent API machine. This folder is automatically granted the necessary Write permissions.

You can edit the registry to specify an alternative file cache location, for example, on a machine with more free disk space. For details, see the EventFilePath description in registry value in Configure the External Agent API.

How much free disk space is required? This depends on: the size of your organization's email archive; the volume of email data removed from your e-mail archive and converted to event files each time you start up the External Agent API (see the previous section); and the file-deletion strategy configured for Event Import.

If you do specify an alternative location, you must ensure that the target folder has Write permissions assigned to the user running the email archive indexer process. If the target folder is on a remote machine, you must specify a UNC path. For example:

```
\\MyMachine\share_name\target_folder
```

Configure the External Agent API

To configure the External Agent API, you need to edit values in the following registry key:

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

Automatically Created Registry Values

Within the **External Agent API** registry key, you may need to edit the following automatically-created registry values:

EventFilePath

Type: REG_SZ

Data: This specifies the full path to the EVF file cache, that is, the folder into which the emails will be written—see the diagram in ZANTAZ EAS integration. This path defaults to the \client\RDIcache folder on the External Agent API machine. You can edit the registry to specify an alternative location, for example, on a machine with more free disk space.

HandledCategory

Type: REG_SZ

Data: (Used for EAS integration only.) Each message in the email archive is assigned to various categories. This registry value appends a category to this list for each message, marking it as 'WGNHandled'. This is to prevent duplicate messages being moved into the EVF file cache. You can edit the registry to rename the default 'WGNHandled' category.

MinFreeDiskMb

Type: REG_DWORD

Data: This specifies a minimum level of free disk space on the machine hosting the EVF file cache—see above. This threshold defaults to 10MB. You can edit the registry to set a different threshold.

If free disk space falls below this level, no further messages are moved from the email archive to the EVF file cache until free disk space increases. Messages still awaiting conversion to event files and subsequent saving to the file cache are flagged accordingly.

As soon as free disk space *does* increase, these flagged messages are picked up when the email archive indexer process (in the case of EAS, this is IndexerService.exe) next runs. They are then removed from the archive, converted to event files and saved to the EVF file cache as normal. No messages are lost from this process because of insufficient free disk space.

RegionTag

Type: REG_SZ

Data: Used by archive integration agents to support regional archives. Change this value only if your organization requires multiple separate archive regions. Specify a short string that uniquely defines the region of the local archive.

Example: Geographic regions, such as "NA", EMEA", "APAC"

Default: Empty String

Note: The RegionTag value specified must match the value defined for the Remote Data Manager (see page 233).

Manually Created Registry Values

In addition to registry values created automatically on installation, the following values are also supported. To use these registry values, create them manually in the **External Agent API** registry key.

DisableNotesAddressLookup

Type: REG_DWORD

Data: Defaults to 0, meaning 'false'. When the External Agent API converts Lotus Notes emails, this controls whether their participants are looked up in the Lotus Notes address books. Set this to a non-zero value (meaning 'yes' or 'true') to disable all address book lookups; set this to zero (meaning 'no' or 'false') to keep the default behavior of participants being looked up in the address books.

0

The External Agent API looks up participants in the address books.

1

Disables all address book lookups.

Note: If DisableNotesAddressLookup is non-zero, i.e. if address book lookups are disabled, the registry value DisableNotesDLExpansion is ignored (if it exists).

DisableNotesDLExpansion

Type: REG_DWORD

Data: Defaults to 0, meaning 'false'. Only used when the External Agent API converts Lotus Notes emails, and if address book lookups are enabled (see the registry value DisableNotesAddressLookup). This option controls whether participants which are Distribution Lists (aka Groups) are expanded into their constituent members. Set this to a non-zero value (meaning 'yes' or 'true') to disable the expansion; set this to zero (meaning 'no' or 'false') to keep the default behaviour of expanding Distribution Lists.

0

Enables the expansion of Distribution Lists (Groups) when the External Agent API converts Lotus Notes emails.

1

Disables the expansion of Distribution Lists (Groups) when the External Agent API converts Lotus Notes emails.

Note: DisableNotesDLExpansion is ignored if the registry value DisableNotesAddressLookup is set, and non-zero, that is, if Lotus Notes address book lookups are disabled. The DisableNotesDLExpansion option is meaningful only when DisableNotesAddressLookup does not exist or is set to zero, that is, if address book lookups are enabled.

EventDateFromSource

Type: REG_DWORD

Data: Defaults to 1. This specifies from where the capture date assigned to imported events is set.

1

The timestamp is set to the date and time in the email. It is based on the delivery time or time sent.

0

The timestamp is set to the date and time when the email was processed by the External Agent.

IgnoreAPMAuditMails

Type: REG_DWORD

Data: Defaults to 0. This specifies whether audit emails are processed by the External Agent API. Set this to a non-zero value (meaning yes or true) to filter out audit emails. Set this to zero (meaning no or false) to make the External Agent API convert audit emails to CA DataMinder events.

0

The External Agent API converts audit emails to CA DataMinder events, this means, they are not 'ignored'.

1

The External Agent API filters out audit emails, this means, they are 'ignored'.

More information:

Overview (see page 148)
EVF File Cache Guidelines (see page 193)

Support for Custom Archives

A 3rd-party integrator uses the External Agent API to receive data from custom archive providers. Up to ten different custom providers are supported, each with a different combination of data type (email or file) and data format (for example, MAPI, RFC2822, files).

Note: For full details, see the *CA DataMinder External Agent COM API Reference Guide*.

More information:

Custom Archive Integration (see page 116)

Chapter 11: Socket API

The Socket API (also known as the Socket agent) enables external applications to use socket connections to call the External Agent API from remote locations, including from non-Windows systems. The Milter MTA agent and the Network Boundary Agent (NBA) both use the Socket API to pass emails and (in the NBA's case) files to policy engines for analysis.

This section focuses on configuring the Socket API. It also includes sections on Socket API throttling and monitoring the Socket API.

This section contains the following topics:

Socket API Requirements

The Socket API requires the Messaging API (MAPI) client libraries to pass email data to a policy engine or PE hub. If this functionality is not available, the Socket API cannot process emails received from the NBA or Milter MTA agent. Therefore, the Socket API host server requires:

MAPI Client and CDO 1.2.1

The Socket API requires Microsoft Exchange Server "Messaging API and Collaboration Data Objects 1.2.1"

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.

Important: Before you install this component, verify that Microsoft Outlook is *not* also installed. If Outlook is already installed, uninstall it.

Note: In previous releases, we recommended 'MAPI client and CDO 1.2.1' *or* Microsoft Outlook. We have now withdrawn this recommendation. Instead, you *must* use 'MAPI client and CDO 1.2.1'.

Install the Socket API

To install the Socket API

Do one of the following:

- Install the Socket API as an option when you install the External Agent API.

  See step 6 of Install the External Agent API.

- Install it as a subfeature when you install a policy engine.

Configure the Socket API

The Socket API is designed to work automatically, without requiring any post-installation configuration. But if you do need to configure the Socket API (for example, to change the default behavior or settings), you need to edit values in the following registry key:

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

The **Socket API** registry key contains the following registry values:

```
AgentPort
AgentPortInterlaced
CreateEAReq
CreateEARsp
DiagnosticFolder
DiskSpaceMinimum
DiskSpaceThrottlingThreshold
HostInHub
HostInPE
ImportQueueLimit
LogLevel
LogMaxNumFiles
LogMaxSizeBytes
MemoryThrottlingThreshold
MemoryUsageLimit
MsgTimeout
UpdateConfig
```

Within the **Socket API** registry key, the **Notifications** subkey contains these registry values:

```
AttachOriginalEmail
AuthType
NotificationFromAddress
SmtpDNSHostName
SmtpServer
UserID
```

These registry values are described in the following sections.

Socket API Registry Key

The **Socket API** registry key contains these values:

AgentPort

Type: REG_DWORD

Data: Defaults to 8538. This registry value specifies which port number the Socket API listens on for data sent by applications using the **Contiguous** protocol. The Milter MTA agent uses this protocol and hence this port. If this port is not used, you can set this value to zero to conserve machine resources.

AgentPortInterlaced

Type: REG_DWORD

Data: Defaults to 8539. This registry value specifies which port number the Socket API listens on for data sent by applications using the **Interlaced** protocol. The Network Boundary Agent uses this port and hence this port. If this port is not used, you can set this value to zero to conserve machine resources.

CreateEAReq

Type: REG_DWORD

Data: Specifies whether to generate External Agent Request diagnostic messages in the folder specified by DiagnosticFolder (see below). If set to:

0 Messages are never generated.

1 Messages are always generated.

2 Messages are only generated on error.

CreateEARsp

Type: REG_DWORD

Data: Specifies whether to generate External Agent Response diagnostic messages in the folder specified by DiagnosticFolder (see below). If set to:

0 Messages are never generated.

1 Messages are always generated.

2 Messages are only generated on error.

DiagnosticFolder

Type: REG_SZ

Data: Specifies the path and folder where External Agent Request and Response diagnostic messages will be saved. The creation of these messages is determined by the CreateEARsp and CreateEAReq registry values.

Note: This folder is not created automatically.

DiskSpaceMinimum

Type: REG_DWORD

Data: Defaults to 800. This registry value specifies the minimum level of free disk space (in MB) on the on the Socket API host machine. If free disk space falls to this level, and while it remains below this level, the Socket API rejects all further messages passed to it.

Note: This registry value refers to free disk space on the drive hosting Windows temporary files.

DiskSpaceThrottlingThreshold

Type: REG_DWORD

Data: Defaults to 1024. This registry value specifies the level of free disk space (in MB) on the Socket API host machine that triggers a change in how messages are processed.

If free disk space falls below this threshold, the Socket API applies 'wait throttling' or 'fail throttling', depending on whether messages were captured by a Milter MTA agent or the NBA.

Note: This registry value refers to free disk space on the drive hosting Windows temporary files.

HostInHub

Type: REG_DWORD

Data: Defaults to 1, indicating the Socket API is connecting to a local PE hub. You do not need to change this registry value.

This registry value is only created on machines hosting the External Agent, Socket API and a Remote Policy Engine Connector (that is, a hub).

HostInPE

Type: REG_DWORD

Data: Defaults to 1, indicating the Socket API is connecting to a local policy engine. You do not need to change this registry value.

This registry value is only created on machines hosting a policy engine and the Socket API.

ImportQueueLimit

Type: REG_DWORD

Data: Defaults to 60. Specifies the maximum number of items that can be queued by the Socket API while they await processing by a policy engine.

We recommend that you set this value to 12 times the total concurrency for all policy engines (if using a hub) or simply 12 times the concurrency of the local policy engine.

Policy engine concurrency is defined by the **Maximum Number of Concurrent Operations** setting in each PE's machine policy. This policy setting sets the maximum number of events that can be processed simultaneously by a policy engine. It prevents a performance slowdown if a policy engine is heavily loaded.

LogLevel

Type: REG_DWORD

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

Log entries are written to WgnSAgent_<date>.log where <date> is the date and time when the log file was created. 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 below), the oldest log file is deleted to enable a new one to be created.

LogMaxSizeBytes

Type: REG_DWORD

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 email server agent creates a new log file. Log entries are written to WgnSAgent_<date>.log; see above.

MemoryThrottlingThreshold

Type: REG_DWORD

Data: Defaults to 300. This registry value specifies the level of memory usage (in MB) on the Socket API host machine that triggers a change in how messages are processed.

If memory usage rises above this threshold, the Socket API applies 'wait throttling' or 'fail throttling', depending on whether messages were captured by a Milter MTA agent or the NBA.

MemoryUsageLimit

Type: REG_DWORD

Data: Defaults to 400. This registry value specifies the maximum level of memory (in MB) used by the Socket API. If memory usage rises to this level, and while it remains above this level, the Socket API rejects all further messages passed to it.

MsgTimeout

Type: REG_DWORD

Data: Defaults to 180. This registry value specifies how long the Socket API waits (in seconds) before rejecting a message. This timeout applies to:

- 'Multipart messages' passed sent to the Socket API. These are emails that have been subdivided into smaller parts before being sent across the network. If the interval between these parts exceeds the specified timeout, the Socket API rejects the email.
- Emails being processed by a policy engine. If the interval between the Socket API submitting an email for policy processing and the policy engine's response exceeds the specified timeout, the Socket API rejects the email.

We recommend that you do not shorten this timeout (for example, do not reduce it to less than three minutes) because this may adversely affect performance.

Note: The Socket API checks an email's progress (in terms of policy processing or the arrival of its constituent parts) every 10 seconds.

UpdateConfig

Type: REG_DWORD

Data: Defaults to 0. Enables administrators to update the Socket API configuration. Set to 1 to force the Socket API to re-read the registry. When it has accepted the changes, it automatically resets this value to 0.

If a registry value has been updated with invalid data, UpdateConfig is set to 2; the new value is discarded and an entry written to the log.

More information:

Socket API Throttling (see page 208)
Monitoring the Socket API (see page 209)

Notifications Registry Subkey

Note: Applicable only to emails captured by the NBA and Milter MTA agent.

For emails captured by the NBA or Milter MTA agent, you can set up the Socket API to allow policy engines to send email notifications to users when their emails are blocked or trigger a warning. To do this, you need to edit values in the following registry key:

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

The **Notifications** registry subkey contains these values:

AttachOriginalEmail

Type: REG_DWORD

Data: Defaults to 1. Specifies whether to attach the original email address to the notification email. If set to 1, the original email **is** attached; if set to zero, then it is **not** attached.

AuthType

Type: REG_SZ

Data: Defaults to None. This specifies which standard SMTP authentication type the Socket API uses to connect to the SMTP server. The following values are supported:

None

Plain

Login

NTLM

CRAM-MD5

We recommend you choose None for unauthenticated connections. However, your SMTP server must be configured to accept connections from the Socket API host machine.

We do not normally recommend Plain or Login authentication because under these protocols the logon password is sent as unencrypted plain text across the network.

NTLM and CRAM-MD5 authentication can be used to connect to SMTP servers on Windows and UNIX machines respectively. However, although these protocols do not send unencrypted logon credentials, you must still ensure that these credentials are protected.

If you use Plain, Login, NTLM or CRAM-MD5 authentication, you must also set up the UserID registry value—see UserID—to pass the logon account details to the SMTP server.

NotificationFromAddress

Type: REG_SZ

Data: Specifies the sender's address that is shown in the From: field of an email notification. For example, you can set this to:

ComplianceTeam@unipraxis.com

SmtpDNSHostName

Type: REG_SZ

Data: This registry value is used to ensure that Socket API notification emails are not reprocessed needlessly by consecutive CA DataMinder email agents.

SmtpDNSHostName specifies a single DNS domain that is written to the email when it is generated by the Socket API. To use this registry value as intended:

1. Set it to the same value (for example, UNIPRAXIS.COM) for all your email server agents.
2. Include this value in the EnterpriseDNSList domain list for Exchange or Domino server agents, or the Milter MTA agent, the parameter enterprisedns-list=<domain list>.
3. When any CA DataMinder server agent receives an email from the Socket API tagged as coming from UNIPRAXIS.COM, it knows that policy does not need to applied to this notification email and so does not process it.

For examples of how the equivalent registry value is used to prevent 'repeat processing' for the Exchange or Domino server agent, see Prevent repeat processing by server agents in multiple domains.

SmtpServer

Type: REG_SZ

Data: Specifies the name of the server hosting the SMTP service and, optionally, the SMTP port number. For example, you can set this value to your Exchange server (if it is configured to relay SMTP messages).

This registry value can also specify the TCP port used for communication between the Socket API and the SMTP server. If omitted, the port number defaults to 25. To specify a non-default port number, append the number to the server name, separated by a colon. For example:

```
unipraxis.com:25777
```

Important! If you change this registry value, you must restart the Socket API host machine for this change to take effect.

UserID

Type: REG_SZ

Data: Specifies a valid user account that the Socket API will use to log on to the SMTP server. This registry value is only required if the Socket API authentication method is **not** 'None' (this is specified by AuthType; see above).

If you need to specify the UserID registry value, you must also securely cache the password for this account. This password will be passed to the SMTP server with the user account name, if required by the authentication type. To cache the password, you run the WgnCred.exe utility on the Socket API host machine, where the component ID is:

EANotifications

You will need to supply this component ID as the <component identifier> if you run a WgnCred.exe command to set the full credentials.

Socket API Throttling

Socket API registry values specify throttling thresholds based on free disk space and memory usage on the host machine. If free disk space falls below the minimum level, or if memory usage rises above the maximum level, this triggers a change in how the Socket API processes messages or files.

For messages captured by an Milter MTA agent, the Socket API applies 'wait throttling'; for messages captured by the NBA, the Socket API applies 'fail throttling'. See below for details.

Wait Throttling for Data From MIlter MTA

(Applies to data sent by Milter MTA agents) For messages captured as they transit through Sendmail or Postfix, the Socket API applies 'wait throttling' if free disk space or memory usage exceed their respective throttling thresholds. This means that, in effect, the Socket API queues messages arriving from a Milter MTA agent and processes them one at a time.

Note: The Milter MTA agent connects to the Socket API on the Agent Port using the **Contiguous** protocol.

Fail Throttling for Data From CA DataMinder Network

(Applies to data sent by CA DataMinder Network) For messages and files captured by CA DataMinder Network as they enter or leave the corporate network, the Socket API applies 'fail throttling' if free disk space or memory usage exceed their respective throttling thresholds. This means the Socket API immediately rejects any items currently being processed and all subsequent items; CA DataMinder Network then permits them to continue, unprocessed.

Note: CA DataMinder Network connects to the Socket API on the Agent Port Interlaced using the **Interlaced** protocol. For full details about CA DataMinder Network, see the *CA DataMinder Network Implementation Guide*.

Monitoring the Socket API

There are various sources of diagnostic information when monitoring the Socket API. These are performance counters, log files and diagnostic files.

Log Files

Socket API log files record the results of message processing.

The file names take the format: WgnSAgent_200808150945.log

The log files are saved on the Socket API host machine in the CA DataMinder \data\log subfolder of the Windows All Users profile.

Socket API Performance Counters

The Socket API includes various the following performance objects that are useful when diagnosing socket connection problems:

- **CA DataMinder External Socket Agent Summary:** Contains counters summarizing overall performance, such as the items successfully analyzed, items currently being processed and failed items.
- **CA DataMinder External Socket Agent Data:** Contains detailed counters for data being processed, such as average processing time, items queued for processing, multipart messages, and various throttling counters.
- **CA DataMinder External Socket Agent Communications:** Contains counters for active connections, plus messages and data received and sent.

Note: For full deployment details, see the *Performance Counters Technical Note*, available from CA Support.

Diagnostic Files

External Agent Request and Response diagnostic messages can be saved in a diagnostic file. For details, see the DiagnosticFolder registry value in Socket API registry key.

More information:

Socket API Registry Key (see page 201)

Chapter 12: ICAP Agent

This section contains the following topics:

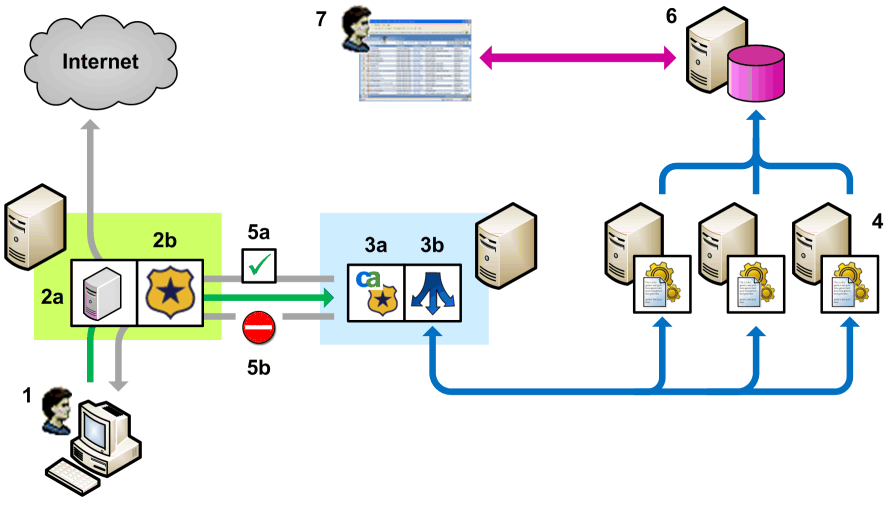
Overview

The ICAP Agent enables CA DataMinder to integrate with Internet Content Adaptation Protocol (ICAP) clients. This provides CA DataMinder with a further method for controlling HTTP activity such as file uploads and downloads.

Organizations run ICAP clients on proxy servers such as Blue Coat ProxySG and Squid to intercept and offload requests initiated from a browser and the corresponding responses from a Web site.

When the ICAP Agent (technically an ICAP server) receives requests from an ICAP clients, it routes them to CA DataMinder policy engines which can then apply Data In Motion triggers, for example, to block inappropriate uploads.

The diagram below shows an example deployment architecture for the ICAP agent and the information flow for an HTTP request.



ICAP agent example deployment architecture

1. A user attempts to upload a file using HTTP. For example, while using a Webmail application, the user attaches a file to the Webmail.
2. **Proxy server and ICAP client:** The email is sent using HTTP or HTTPS to the proxy server (**2a**). The ICAP client (**2b**) on the proxy server intercepts the request and routes the file to the ICAP agent (**3a**).
3. **ICAP agent and hub:** The ICAP agent (**3a**) passes the file to a Remote PE Connector (**3b**), which in turn allocates it a policy engine (**4**).
4. **Policy engines:** A PE analyzes the file. The outcome of any policy processing ('block' or 'allow') is routed back via the ICAP agent to the ICAP client.
5. **Result of policy processing:** These results are routed back to the ICAP client. If the result is:
   - 'Allow' (**5a**), the upload is permitted and the request is processed by the ICAP client.
   - 'Block' (**5b**), the upload is blocked and the ICAP client routes a notification message to the user's browser.
6. **CMS:** Any resulting events are replicated up to the CMS and stored for subsequent retrieval and reviewing (**7**).

Integration Procedure for ICAP Clients

Integrating CA DataMinder with an ICAP client involves the following tasks:

1. (Optional) Import user DN details.
2. Install the ICAP agent and policy engine hub.
3. Deploy one or more policy engines.
4. Configure the ICAP agent and policy engine hub.
5. Configure your proxy server and ICAP client.

Import DN Details to CA DataMinder User Address Lists

(Applies specifically to integration with BlueCoat ProxySG servers)

To prevent potential policy processing delays, we recommend that you import user DN details ('distinguished names') from your LDAP directory and add them to the email address lists for your CA DataMinder users. This enables policy engines to apply the correct user policy to HTTP activity without needing to perform an LDAP directory lookup.

To add DN details to user address lists, you must run an Account Import job.

Use the Account Import wizard

Set up your account import job as normal, but in the Email Attributes screen (step 9):

1. Clear the 'Use default attributes' check box.
2. Select any available attribute and click 'Add'.
3. In the resulting Edit Selection dialog, type **distinguishedName** and click Add.

From a command line

Define your wgninfra account import command as normal, but add the following parameter:

```
/ml distinguishedName
```

Note: For details about command line operations and import parameters, see the Account Import chapter in the *Platform Deployment Guide*.

Install the ICAP Agent

You install the ICAP Agent using the CA DataMinder Integration Agents installation wizard.

To install the ICAP 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 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 ICAP Agent.

   A policy engine hub is installed automatically with this 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.

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.

Configure the ICAP Agent

The ICAP agent is configured to work automatically with Blue Coat ProxySG ICAP clients. But if you are using a different proxy server or if you use an alternate configuration, you need to configure the ICAP agent. To do this, edit values in this registry key:

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates
    \CA DataMinder\CurrentVersion\ICAP
```

This registry key contains the following registry values:

AgentPort

Type: REG_DWORD

Data: Defaults to 1344. This registry value specifies the port used by the ICAP client and server (that is, the ICAP agent) to communicate ICAP requests and responses.

Note: If you change the port value, you must restart the PE hub.

AuthenticatedUserEncoding

Type: REG_DZ

Data: Defaults to ‘auto’. This registry value specifies the encoding scheme for the user name in the ICAP message.

Supported values are auto, base64, and none:

auto

The ICAP Agent automatically tries to detect whether the authenticated user information is base64 encoded. Choose the 'auto' option in situations where the user information passed to the ICAP agent can be either base64 encoded or plain text.

If the ICAP agent confirms that:

- The user information *does* use base64 encoding, the agent decodes the user information before parsing it.
- The user information does *not* use base64 encoding, the agent infers that the user information is in plain text. The agent therefore parses the user information without decoding it.

Important! The 'auto' option does not work if the authenticated user name has an ambiguous format.

User names with ambiguous formats

If the agent receives user information that contains characters that cannot occur in valid base64 encoding (such as colons and backslashes), the agent immediately confirms that the user information is *not* base64 encoded.

However, if the user information contains only characters that are valid in base64 encoding, the format is ambiguous. The ICAP agent cannot determine whether the user information is base64 encoded or in plain text and so assumes that it is base64 encoded. Valid characters in base64 encoding are limited to letters, digits, and '+' and '/' characters.

In particular, if user information is passed to the ICAP agent in *domain/username* format, the 'auto' option does not work. For example, the following user name is valid as both plain text *and* base64 encoded:

```
"unipraxis/srimmel"
```

Conversely, the 'auto' option does work correctly if the ICAP agent receives user information that uses the following formats:

```
"LDAP://10.0.8.50/CN=Spencer Rimmel,CN=Users,DC=unipraxis,DC=com"
"WinNT://unipraxis/srimmel"
"unknown://unipraxis\srimmel"
"unknown://srimmel@unipraxis.com"
"unknown://CN=Spencer Rimmel,CN=Users,DC=unipraxis,DC=com"
"srimmel@unipraxis.com"
"unipraxis\srimmel"
"CN=Spencer Rimmel,CN=Users,DC=unipraxis,DC=com"
```

Note: If you see multiple E3F23 errors in the ICAP agent log, the ICAP agent is failing to correctly parse user information in plain text. You must therefore change AuthenticatedUserEncoding to 'none'.

base64

Specifies a base64 encoding scheme on the proxy server. Choose this option if the user information passed to the ICAP agent is always base64 encoded.

none

Choose this option if the user information passed to the ICAP agent is always in plain text.

AuthenticatedUserHeader

Type: REG_DZ

Data: Defaults to X-Authenticated-User. This value is the default for Blue Coat ProxySG servers.

This registry value specifies the ICAP x-header that contains the user credentials. Policy engines use these credentials to map the user to a CA DataMinder user account. If you use a different proxy server, you must set this value to identify the ‘user credentials’ header used by that proxy server.

AuthenticatedUserType

Type: REG_DZ

Data: Defaults to auto. This registry value specifies what type of user information is included in the AuthenticatedUserHeader x-header. Policy engines use this 'user type' information to determine the user policy to use when processing the data. Supported values are auto, DN, user, and SMTP:

auto

The ICAP agent tries to detect the format automatically and extract the user information. The agent can detect distinguished names, 'domain\user' names, and SMTP email addresses.

The agent detects user information prefixed with any of the following Blue Coat ProxySG prefixes: LDAP, WinNT, and 'unknown'. For example:

```
LDAP://10.0.8.50/CN=Spencer Rimmel,CN=Users,DC=rimmel,DC=com
WinNT://unipraxis/srimmel
unknown://srimmel@unipraxis.com
```

The agent also detects user information without the prefixes listed above. For example:

```
10.0.8.50/CN=Spencer Rimmel,CN=Users,DC=rimmel,DC=com
unipraxis/srimmel
srimmel@unipraxis.com
```

DN

DN is for Blue Coat ProxySG servers that use LDAP authentication. DN indicates that AuthenticatedUserHeader is populated with the user's DN entry in the LDAP directory. For example:

```
LDAP://10.0.8.50/CN=Spencer Rimmel,CN=Users,DC=rimmel,DC=com
10.0.8.50/CN=Spencer Rimmel,CN=Users,DC=rimmel,DC=com
unknown://CN=Spencer Rimmel,CN=Users,DC=rimmel,DC=com
CN=Spencer Rimmel,CN=Users,DC=rimmel,DC=com
```

user

user is for Blue Coat ProxySG servers that populate the AuthenticatedUserHeader with prefixed 'domain\user' user credentials. The Blue Coat IWA and Windows SSO authentication methods generate these credentials. For example:

```
unknown://unipraxis\srimmel
unknown://unipraxis/srimmel
unipraxis\srimmel
unipraxis/srimmel
```

SMTP

SMTP is for Blue Coat ProxySG servers that populate the AuthenticatedUserHeader with prefixed SMTP email addressesThe Blue Coat Policy Substitution authentication method generates these addresses. For example:

```
unknown://srimmel@unipraxis.com
srimmel@unipraxis.com
```

CreateICAPMsg

Type: REG_DWORD

Data: Defaults to 2. This specifies how ICAP messages passed to the ICAP agent are stored in the specified DiagnosticFolder (see below). Supported values are:

0 Do not write any messages to the diagnostic folder.

1 Dump every message to the diagnostic folder. Only use this value if directed to do so by CA Support.

2 Only write messages to the diagnostic folder if a processing error occurs.

If no diagnostic folder is specified (that is, if DiagnosticFolder is blank), no ICAP messages are written, regardless of what value CreateICAPMsg is set to.

ClientIPHeader

Type: REG_DZ

Data: Defaults to X-Client-IP, the default value for Blue Coat ProxySG servers. This registry key value specifies the portion of the HTTP header that contains the IP address of the user system. If using a different proxy server, you may need to modify this value.

DiagnosticFolder

Type: REG_DZ

Data: No default value. This specifies the folder into which ICAP messages passed to the ICAP agent are saved for diagnostic purposes. The level of saved messages is set by CreateICAPMsg (see above).

You may be asked to modify this value by CA Support.

HostInHub

Type: REG_DWORD

Data: Defaults to 1. This determines whether the ICAP agent connects to a local policy engine hub. Do not change this value.

LogLevel

Type: REG_DWORD

Data: Defaults to 2. This determines the level of logging for the ICAP server. For example, you can configure the ICAP server to only log errors or important system messages.

Log entries are written to the WgnICAP_<date>.log file, where <date> is the date and time when the log file was created; the file is located in CA DataMinder's \data\log subfolder of the Windows All Users profile; see Viewing log files. The supported logging levels are:

1 Errors only

2 Errors and warnings

3 Errors and warnings, 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 and diagnostic purposes only. For example, it shows storage and retrieval on every resource item.

MaxMessageSizeMB

Type: REG_DWORD

Data: Defaults to 50. This specifies the maximum size (in MB) of message to be processed by the ICAP agent. Messages larger than this are allowed, but not processed; an entry is written to the log file indicating that a message exceeded the maximum size and was allowed.

To avoid unnecessary delays, ensure that this maximum file size threshold matches or is less than the Maximum Size of Files (KB) setting in the user policy (which also defaults to 50 MB). This prevents very large files being sent for policy processing by the ICAP agent, only to be rejected if they exceed the user policy-defined threshold.

ResponseTemplateFile

Type: REG_DZ

Data: Specifies the name and path to the HTML template file that contains the notification message shown to users for HTTP responses from Web sites as a result of policy processing. This registry value defaults to:

C:\Program Files\CA\CA DataMinder\client\ResponseTemplate.html

This default template is very generic. It contains variables for the message text issued by the policy engine. If required, you can modify the template content and the file name and location to meet your organization’s needs.

RequestTemplateFile

Type: REG_DZ

Data: Specifies the HTML template file that contains the notification message shown to users for HTTP requests as a result of policy processing. This registry value defaults to:

C:\Program Files\CA\CA DataMinder\client\RequestTemplate.html

This default template is very generic. It contains variables for the message text issued by the policy engine. If required, you can modify the template content and the file name and location to meet your organization's needs.

UpdateConfig

Type: REG_DWORD

Data: Defaults to 0. Enables administrators to update the ICAP agent configuration. Set to 1 to force the ICAP agent to re-read the registry. When the ICAP agent has accepted the changes, it automatically resets this value to 0.

Configure the Proxy Server and ICAP Client

The proxy server must have an ICAP client installed. ICAP clients can process both requests and responses. See the documentation for your proxy server and ICAP client for full configuration details. For integration with CA DataMinder, you must:

1. Verify that a supported authentication method is enabled on the proxy server.

   If authentication is not enabled, the user identity cannot be passed from the ICAP client to the ICAP agent.

   If you are using a Blue Coat ProxySG server, see your Blue Coat documentation for details about configuring the LDAP, IWA, Windows SSO or Policy Substitution authentication methods.

   Note: If you use a Blue Coat ProxySG server and single sign-on authentication, verify that the sso.ini file identifies the user account that the BlueCoat service runs as. The sso.ini file specifies this user account in the '[SSOServiceUsers]' section.

2. Specify the port number assigned to the ICAP agent.

   This port must match the port specified by the AgentPort registry value. By default, this is 1344.

3. Identify the ICAP agent host machine. This allows the ICAP client to route requests and responses to the ICAP agent.

   For example, for Blue Coat ProxySG servers you must provide a service URL, formatted as shown below:

   `icap://<ipaddress>:<port>/<reqmod>|<respmod>`

   where:

   <ipaddress> is the IP address of the ICAP agent host machine.

   <port> specifies the port number used for communication between the ICAP client and ICAP agent. If you use the default port (1344), you can omit the port number from the URL.

   <reqmod>|<respmod> identifies the type of event, an HTTP request or HTTP response.

4. Enable the ICAP v1 options Client Address and Authenticated User.

5. Disable Preview Size for HTTP responses.

   This option is not supported by the CA DataMinder ICAP agent. If it is enabled, the ICAP agent may not work correctly.

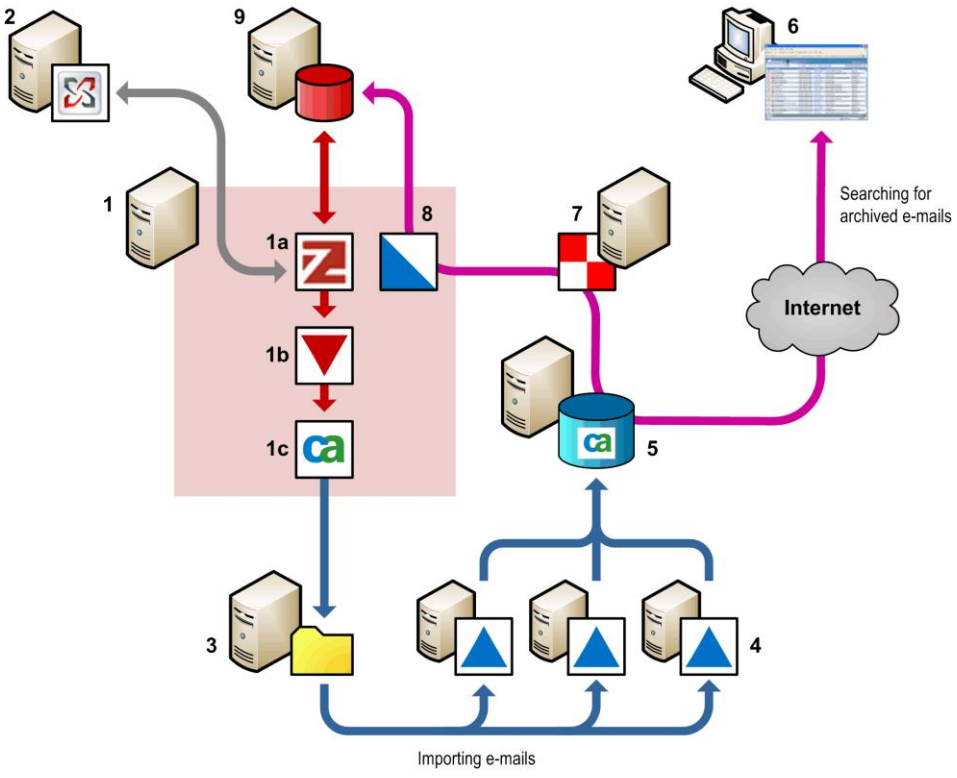
Chapter 13: Remote Data Manager

This section contains the following topics:

Remote Data Manager

The Remote Data Manager (RDM), Wgnrdm.dll, enables CA DataMinder to retrieve events that are archived in third party remote storage locations and display them in the iConsole or Data Management console. The RDM submits data requests through a Microsoft Internet Information Services (IIS) server.

The example below shows how the RDM (**7**) is used to integrate with ZANTAZ EAS.



RDM and integration with ZANTAZ EAS

1a ZANTAZ EAS
1b EAS IndexerService.exe
1c External Agent API

2 Microsoft Exchange

3 EVF file cache

4 Event Import utility

5 CMS

6 iConsole

7 RDM. When displaying captured emails in the iConsole (**6**), the RDM retrieves data for emails archived in the e-mail store (**9**). These requests are sent via IIS (**8**).

8 Microsoft IIS

9 E-mail store

Remote Data Manager Support for Custom Archives

If you are integrating with a custom archive developed by a 3rd party integrator, configure RDM to connect to that archive to retrieve historic emails and files. RDM uses HTTP GET requests to retrieve data from custom archives.

More information:

Custom Archive Integration (see page 116)

RDM Requirements

The following are RDM requirements:

Archive solutions

For the current release, the RDM utility supports Zantaz EAS, Symantec Enterprise Vault, and EMC SourceOne.

Note: RDM configuration for Enterprise Vault requires the RDM to be installed on the SEV server or EV Administration Console.

Email solution

The host machine must have an e-mail client installed:

- Outlook 2003 or later if retrieving Exchange e-mails from a third party archive.
- Lotus Notes 6.5 or later if retrieving Domino e-mails from a third party archive.

Important! This must be the default email application on the host machine.

CA DataMinder infrastructure

You must change the logon account for the infrastructure service to a named user. For integrations with EAS, EV and SourceOne, there are additional requirements.

IIS server

RDM requests are serviced by an IIS server; you must specify the IIS server by name, not by IP address. For details, see step 6 in 'Install the RDM'.

Install the RDM

You install the RDM using the CA DataMinder server installation wizard. But note the requirements above.

To install the RDM

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 CA DataMinder Platform and then click Install.

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

4. In the server installation wizard, navigate to the Customer Setup screen.

5. In the Custom Setup screen, expand the Enterprise Server feature and choose Remote Data Manager.

6. In the Service Accounts screen, specify the logon account used by the local CA DataMinder infrastructure service, wgninfra.exe.

   Important! By default, the infrastructure logs on as a LocalSystem account but you must change the service's logon account to a named user!

   Click the Browse button  for the Infrastructure service, then enter the domain, name and password for the account you want to use. This account requires the 'Log on as a service' security privilege. Note also:

   ZANTAZ EAS integration

   The CA DataMinder infrastructure service account must have EAS permissions to retrieve archived data. In EAS, this account, must be listed in the Account Administrator dialog *and* must have the IIS Content Retrieval for Indexing\Restore permission. This prevents authentication errors when the IIS server connects to the email archive server.

   Symantec Enterprise Vault integration

   The CA DataMinder infrastructure service account must be the same as the EV 'Service User'. For details, refer to your Enterprise Vault documentation.

   EMC SourceOne integration

   The CA DataMinder infrastructure service account must must be assigned Administrator permissions to all mapped folders on which archived messages are stored.

   Custom Archive Integration

   Use a Windows account with access to the custom archive as specified by the 3rd-party integrator.

7. In the Remote Data Manager Configuration screen, specify the target archive.

   Archive type

   Choose the archive that you want to integrate with.

   Note: For custom archives, you perform all configuration after the installation.

   EAS Server and Port

   Only applicable to EAS integration. Specify the name of the IIS server that will service RDM requests. You must also set the port number on which IIS listens for data requests from the RDM. This defaults to port 80.

   Important! You must specify the IIS server by name, not by its IP address, even if the IIS server and RDM are on separate subnets. If you specify its IP address, the IIS server will not be able to authenticate the RDM.

SourceOne Server

(For EMC SourceOne integration only) Specify the name or IP address of the machine hosting SourceOne Document Management Service.

Note: If you change the SourceOne Document Management Service host after you install the RDM, edit the DMSServer registry value to specify the new host machine.

8. In the final wizard screen, click Install to start the file transfer.
9. Perform any necessary post-installation tasks. In particular, ensure that you have:
   - Assigned the 'Log on as a service' security privilege' to the infrastructure logon account.
   - Included the EAS server as a trusted intranet site.

   Further post-installation tasks are required to enable the RDM to retrieve events from the following archives and applications:
   - IBM Content Collector (see page 229)
   - EAS integration (see page 229)
   - EMC SourceOne integration (see page 231)
10. Restart the local CA DataMinder infrastructure service to activate the RDM.

More information:

Assign the 'Log on as a service' privilege (see page 228)

RDM Post-installation Tasks

After installing the RDM, you may need to perform further tasks.

More information:

Assign the 'Log on as a service' privilege (see page 228)
Configure the File Retrieval Timeout (see page 228)
IBM Content Collector Integration (see page 229)
EAS Integration (see page 229)
Enterprize Vault Integration (see page 230)
EMC SourceOne Integration (see page 231)
Configure Custom Archives (RDM) (see page 231)
Define the Archive Region (see page 233)

Assign the 'Log on as a service' privilege

In step 5 of Install the RDM, you changed the logon account used by the local CA DataMinder infrastructure service, Wgninfra.exe, from LocalSystem to a named user. This account requires the ‘Log on as a service’ security privilege. If you have not already done so, you must assign this privilege manually.

1. Ensure that you are logged on with local administrator rights on the RDM host machine.
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 service’ privilege to the CA DataMinder infrastructure logon account.

Configure the File Retrieval Timeout

You can configure the maximum period of time that CA DataMinder will allow for the RDM to fetch a file from the relevant data source. You may need to extend the time if very large files need to be retrieved, or if a data source has limited bandwidth. Locate this registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
  \CurrentVersion\Remote Data Manager
```

Within this registry key, you need to configure the following registry value:

GlobalFileRetrieveTimeoutMilliseconds

Type: REG_DWORD

Data: Defaults to 300,000 (equivalent to 5 minutes). This specifies the file retrieval timeout in milliseconds. If the RDM does not respond within this period, the file retrieval process is abandoned.

IBM Content Collector Integration

To ensure that the RDM can retrieve events from Content Collector, you must specify the Content Collector host server. To do this, locate this registry key on the RDM host server:

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

Within this registry key, configure the following registry value:

URL

Type: REG_SZ

Data: Enter the following information:

`https://<server:port>/DocViewer/`

Where *<server:port>* specifies the host server and the port number used by the CA DataMinder web service. For example:

`https://<UX-CC-ARCHIVE-NY:11443>/DocViewer/`

EAS Integration

To ensure that the RDM data requests can be authenticated by the IIS, you need to add the EAS server to the local intranet's list of trusted sites. To do this:

1. On the machine hosting the RDM server, edit the Internet Explorer Security options for the local intranet.
2. Add the URL for the EAS server to the list of sites specified for the local intranet zone. This must be the same name that you specified in step 5 of installing RDM. For example:

   `http://<UX-MAIL-ARCHIVE-NY>`

   Note: You must be logged into Windows using the same account as the named user the RDM uses on the infrastructure.

Enterprize Vault Integration

(Applies to RDM searches for archived Domino emails only)

When the RDM retrieves events from an Enterprise Vault archive, it performs a *federated search*. By default, a federated search runs across all index volumes concurrently, *up to a maximum of five volumes*. If an archive contains more than five index volumes, the RDM runs a *non-federated search*. A non-federated search runs on one volume at a time and stops when it finds the specifed item.

For archives with a large number of volumes, a non-federated search is considered more efficient. However, if you want the RDM to always run federated searches but your archive has more than five index volumes, you can increase the maximum number of index volumes that can be searched concurrently. Locate this registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
    \CurrentVersion\Remote Data Manager
```

Within this registry key, configure the following registry value:

MaxFedSearchVolumes

Type: REG_DWORD

Data: Defaults to 5. This value defines the maximum number of index volumes that the RDM can search concurrently as a federated search. If the number of volumes exceeds this value, the RDM runs a non-federated search.

EMC SourceOne Integration

1. If your RDM is installed on a remote server (that is, it is **not** on the SourceOne management server), you must install the EMC SourceOne SRE Initialization package (ES1_SRESetup.exe) on the RDM host server. This enables the RDM to connect to SourceOne.

   Note: The SRE Initialization package is supplied by EMC.

2. This component also requires a Notes client with specific rights and a suitable password:

   a. Install a Notes client. The default user for this client must have sufficient rights to create a Notes database on the RDM server.

   b. Create a password for the Notes client account and store it in the registry. From a command prompt in the \system subfolder in the CA DataMinder installation folder, run:

      ```
      wgncred -set
      ```

      This displays a list of components for which passwords are stored plus their corresponding ID numbers and component identifiers.

   c. When prompted, enter the ID number for RDMEELDClient.

   When prompted, enter and confirm your chosen password for the Notes client account.

Configure Custom Archives (RDM)

If you are integrating with a custom archive provider, configure RDM to retrieve the email or file object in the correct format and in the correct location. Make all configuration for custom providers in the registry under the following keys:

- On 32-bit systems:
  `HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder \CurrentVersion\Remote Data Manager`
- On 64-bit systems:
  `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ComputerAssociates\CA DataMinder \CurrentVersion\Remote Data Manager`

Follow these steps:

1. Enable support for the custom archive in this registry key by modifying the AllowCST*xx* registry value with a name based on the numeric value of the importSource used for this custom archive. The third-party integrator who implemented support for the custom archive provides you with this value. You can configure multiple archives.

   Note: See the "Common Parameter Details, Values for importSource" section in the *CA DataMinder External Agent COM API Reference Guide*.

   AllowCST*xx*

   Enables the custom archive that is specified by the importSource value. Replace *xx* with the decimal value of importSource.

   Type: REG_DWORD

   Data: Defaults to 0. Set to 1 to enable support for the custom archive. After changing this value, restart the CA DataMinder Infrastructure service on the RDM machine to ensure this value is registered with the CMS.

2. To configure the connection to the particular archive, locate the registry subkey CST*xx*, where *xx* is the decimal value of importSource.

3. Within the CST*xx* subkey, edit the following registry values:

   Type

   Defines the type of data that the archive returns.

   Type: REG_SZ

   Data: Set to EMAIL or FILE.

   Format

   Defines the format in which the archive returns data.

   Type: REG_SZ

   Data: Set to MAPI, RFC822, or FILE.

   URLTemplate

   Defines the URL to the archive provider.

   Type: REG_SZ

   Data: This value defines the URL template that the RDM uses to request the data from the archive. The URL template encodes the uniqueID of the data in the archive, so it must contain a ‘%s’ format definition. The '%s' format definition is replaced by the RDM when issuing the request to the archive.

   Example: This archive returns emails in RFC822 format:
   `http://server:port/folder/%s.eml`

The following table shows the supported combinations of the Type and Format settings:

Format/Type	EMAIL	FILE
MAPI	Supported	N/A
RFC822	Supported	N/A
RFC2822	Supported	N/A
FILE	N/A	Supported

Note: You do not need to restart the local CA DataMinder Infrastructure service when you modify the settings under the CST*xx* subkey.

Define the Archive Region

CA DataMinder supports regional archives (see page 117). If you have multiple unconnected email archives of the same type, you can use RegionTags to identify them.

Follow these steps:

1. Edit the following registry key:
   `HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder \CurrentVersion\Remote Data Manager`
2. Configure the RegionTag registry setting:

RegionTag

Type: REG_SZ

Data: Used by archive integration agents to support regional archives. Change this value only if your organization requires multiple separate archive regions. Specify a short string that uniquely defines the region of the local archive.

Example: Geographic regions, such as "NA", EMEA", "APAC"

Default: Empty String

Note: The value specified must match the value defined for the archive agent machines (see page 193).

You have configured a region tag for a Remote Data Manager Server.

Do Not Rename Your Archive Servers!

Important! If you rename an archive host server, CA DataMinder cannot retrieve messages stored in the archive!

If you rename an archive host server, any references within CA DataMinder to emails archived on that server will no longer be valid, as the machine name forms part of the unique event identifier. This means that although the CMS will keep a summary of event data for such events, their content will no longer be retrievable.

Support for Multiple RDM Servers

It is possible to have multiple RDMs connected to the same CMS. This enables CA DataMinder to retrieve events from third party remote storage locations more efficiently by sharing requests across all installed RDMs.

If an RDM stops responding, future requests for archived files are directed to any remaining RDM servers. The CMS will try to reconnect to the failed RDM after 15 minutes. To configure this retry interval, you must edit the startup.properties file:

1. Stop the 'CA DataMinder infrastructure' service. From a command prompt, run:

   ```
   net stop wgninfra
   ```

2. Edit the startup.properties file. Find this file in the \system subfolder of the CA DataMinder installation folder.

3. Open this file and add the following line:

   ```
   service.retrydelaysecs=600
   ```

   The retry interval is specified in seconds. This example sets the retry interval to 10 minutes.

4. Restart the CA DataMinder infrastructure service. From a command prompt, run:

   ```
   net start wgninfra
   ```

   Note: For full details, please contact CA Technical Support.

Chapter 14: Universal Adapter

This section contains the following topics:

What is the Universal Adapter?

The Universal Adapter (UA) imports emails from multiple Microsoft Exchange and Domino journal mailboxes and pre processes them before they go into an email archive system. Specifically, it can resolve e-mail duplication, unwrap emails that have been through envelope journaling and perform distribution list expansion, showing the members of any recipient lists at the time the email was sent. It can also optionally integrate with a policy engine to apply policy and add smart tags to each email.

Inputs and Outputs

The Universal Adapter can import emails from multiple Exchange and Domino journal mailboxes. You will create a registry source structure to support this. The Universal Adapter then outputs the imported e-mails to any of the following:

Exchange or Domino mailbox

From the output mailbox, the emails can then be archived by a third party application, or imported into the CMS.

EVF files

These are CA DataMinder event files. They can be imported onto a CMS by the CA DataMinder Event Import utility.

Third party DLL

Microsoft Exchange emails only. The Universal Adapter can output emails to a third party DLL (using a CA-defined interface) for archiving to a third party storage solution.

Again, you need to create a registry source structure to support your outputs.

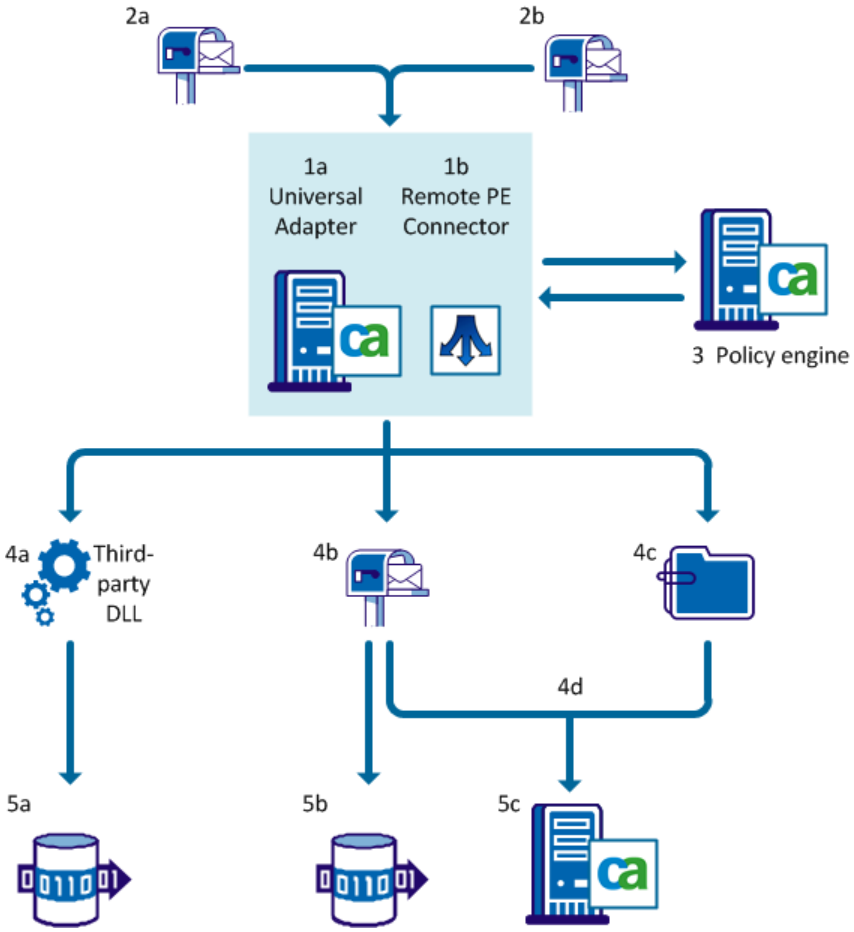
More information

Input Mailbox Overview (see page 257)
Set Up and Configure the Input Source Structure (see page 256)
Set Up the Output Structure (see page 269)

Universal Adapter Architecture

The diagram below shows how the Universal Adapter: imports emails from journal mailboxes; processes them; optionally integrates with policy engines to apply policy and assign smart tags; outputs the imported emails to various locations and then archives them.

Universal Adapter Data Flow



The Universal Adapter Architecture illustration is described as follows:

1. **Universal Adapter.** You can optionally connect the Universal Adapter (**1a**) to a policy engine (**2**), via a hub (**1b**).
2. **Journal Mailboxes.** The Universal Adapter can import emails from multiple **(2a)** Exchange and **(2b)** Domino journal mailboxes.
3. **Policy engine.** You can optionally connect the Universal Adapter to a **(3)** policy engine (via a hub) to apply policy to emails and populate them with smart tags.
4. **Outputs.** The Universal Adapter then outputs the processed emails to: **(4a)** (for Exchange emails only) a third-party DLL; **(4b)** an Exchange or Domino mailbox; **(4c)** EVF files; or **(4d)** Event Import.
5. **Archives.** Finally, the emails are passed to a **(5a)** storage solution, **(5b)** archive or **(5c)** alternatively to Event Import for importing into CA DataMinder.

De-enveloping

(For Microsoft Exchange emails only)

The envelope journaling feature in Microsoft Exchange enables organizations to archive transport envelope data. This identifies the recipients who received the message, including recipients from distribution lists and blind carbon-copy (Bcc) recipients. When an email has been through the envelope journaling process, the result is an envelope message, or 'wrapper' email, that contains a journal report plus an attachment containing the original email. In effect, it reconstitutes a version of the original e-mail. The journal report (the body text of the envelope message) contains the transport envelope data of the original email.

The Universal Adapter can be configured to reformat e-mails that have been through the envelope journaling process. That is, it unwraps them so the envelope data, email message body and the complete list of recipients (represented in their true To, CC, From field allocation) are all available in one flat message format. This is much more convenient for any reviewers who subsequently need to review the email after it has been archived.

Note: The detailed recipient information is presented as a hidden property and must be read by a bespoke application.

Expanding Distribution Lists

When an email is sent to a distribution list (DL), only the DL display name is visible in the To: field of the message, for example To: All Marketing. When processing an email addressed to a DL, the Universal Adapter can optionally expand the DL. That is, it connects to the organization's LDAP directory service to look up the DL members and identify the email addresses of the actual recipients.

If the Universal Adapter is configured to expand the DLs, it stores the individual recipients in the To: and Cc: fields. For example, if the display names are:

To: All Marketing

Cc: Spencer Rimmel

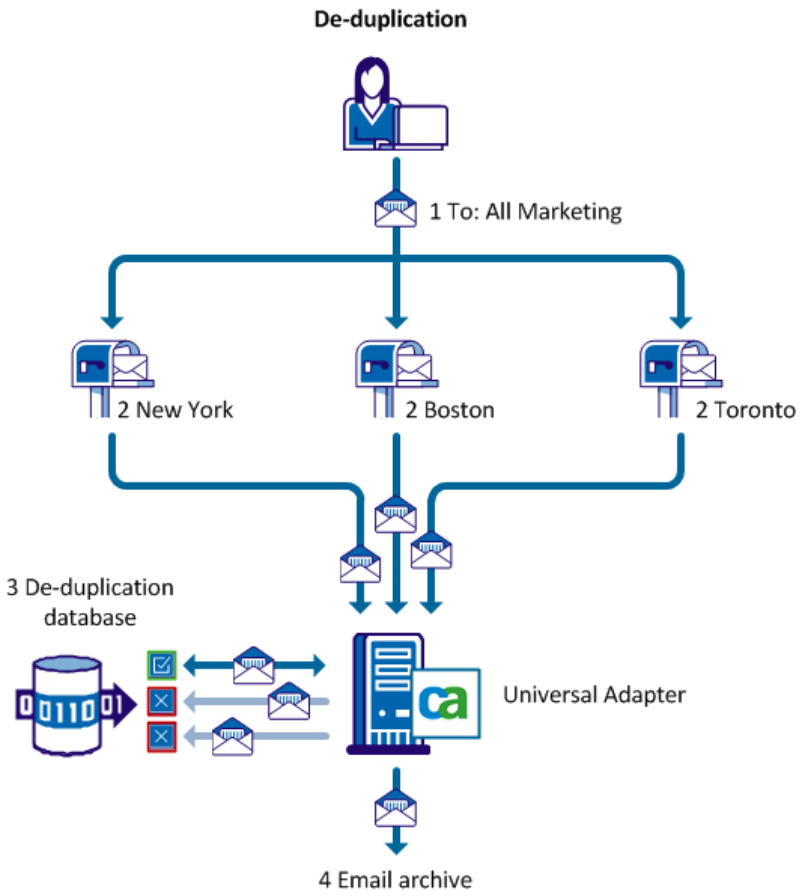
Then the stored recipients are actually:

To: Lynda Steel; Frank Schaeffer; Omar Abassi; Qi Xaopeng

Cc: Spencer Rimmel

De-duplication

When an email is sent to multiple recipients, it is often the case that recipient accounts are hosted across multiple Exchange servers or Domino servers. If, for example, these servers host their own journals, there will then be multiple copies of the original email, one in each journal on the recipient mailbox servers, plus another one in the journal on the sender's mailbox server.



De-duplication email flow:

1. **Single user.** A user sends an email in the New York office to the 'All Marketing' distribution list.
2. **Journal mailboxes (Exchange or Domino).** There are members of the distribution list in the New York, Boston and Toronto offices, each of which runs its own email server with its own journal. A copy of the email now exists on each journal server.

3. **De-duplication database.** The Universal Adapter processes these emails via the de-duplication database, which stores a unique ID for each email. The ID is identical for each duplicate email, enabling the de-duplication database to filter out emails already processed by the Universal Adapter. The New York copy is processed first, and the other two copies are identified as duplicates.
4. **Email Archive.** The Universal Adapter then sends a single copy of the email to its outputs.

In the previous example, an email is sent from a user in the New York office to the All Marketing distribution list. There are members of the distribution list in the New York, Boston and Toronto offices, and each of the offices runs its own Exchange or Domino server with its own journal. A copy of the email now exists on each of these journal Exchange or Domino servers.

The Universal Adapter processes these emails via the de-duplication database, which stores a unique ID for each email. The ID will be identical for each duplicate email, which enables the de-duplication database to filter out emails that have already been processed by the Universal Adapter. In this example, the New York copy is processed first, and the Boston and Toronto copies are identified as duplicates. The Universal Adapter therefore only sends a single copy of the e-mail to its outputs. Without the Universal Adapter, three identical emails would be stored in the e-mail archive, using up valuable storage space.

De-duplication Database Partitions

You can control the period for which unique IDs are held by changing the frequency at which the "UA Partition Mgmt" SQL Server Agent job runs.

The de-duplication database consists of five partitions. The Universal Adapter writes to each of these in turn, switching from one to the next when the "UA Partition Mgmt" SQL Server Agent job runs. After switching to a new partition, the Universal Adapter truncates the partition before writing any new values to it. Therefore, if you change the frequency at which the job runs, you can control how long the Universal Adapter retains unique IDs. For example, setting the job to run every day means that if an email is processed that is a duplicate of an e-mail seen within the last five days, it will be detected as a duplicate and not written to the outputs. Specifically, in this example, data from the last four days is kept plus data that has accrued in the current partition since the last partition switch took place.

Policy Engine Integration

You can optionally link the Universal Adapter to policy engines in order to apply policy and add smart tags to each email. You can then also use the policy engine to generate CA DataMinder email events if required.

Universal Adapter and Smart Tagging Overview

If you connect the Universal Adapter to a policy engine, all emails processed by the Universal Adapter that cause a policy trigger to fire will be assigned a smart tag.

Note: The Universal Adapter connects to policy engines via a policy engine hub; it cannot connect directly to a policy engine.

UA Requirements

Operating System

The Universal Application is included in the UA.msi installation package.

UA.msi supports 32-bit versions of these operating systems:

- Windows Server 2003 (see note 1)
- Windows Server 2008 (see note 1)

UA.msi also supports 64-bit versions of these operating systems:

- Windows Server 2008 R2
- Windows Server 2012

Note 1: We have not tested these operating systems with the current versions of UA.msi.

Email Servers

The UA supports the following email servers:

- Microsoft Exchange Server 2003, 2007 or 2010
- Lotus Domino 7.0.2, or 8

Local Email Client

One of the following email clients must be installed on the target UA server:

- Lotus Notes 7 or 8
- Microsoft Outlook.

  This enables the UA to temporarily store emails in .PST files while they are processed. If you use Outlook as your MAPI client, Outlook *must* be the default email application on the host machine.

  Important! We recommend Outlook 2003 if you are concerned about performance. Our UA testing found that performance with Outlook versions other than 2003 can impact performance. If you do use a later version of Outlook, note that the UA only supports 32-bit versions of Outlook.

Installing the Universal Adapter

This chapter covers how to install a standard Universal Adapter, as well as optional extras such as the de-duplication database and policy engine integration. Installing the Universal Adapter is a multi-step process:

To install Universal Adapter

1. **Set up a Universal Adapter domain user.**

   The Universal Adapter service must run as a domain user who can access the relevant mailboxes hosted on each input and output Exchange server.

2. **Install the Universal Adapter.**

   You need to run the Universal Adapter installation wizard and optionally installing the de-duplication database.

   a. Run the UA installation wizard.

   b. Install the de-duplication database.

3. **Set up policy engine integration.**

   To install one or more policy engines, you need to run the Server installation wizard. To install support for the policy engine hub (via the remote policy engine connector), you need to run the Integration Agents installation wizard.

   a. Install and configure the policy engines.

   b. Install and configure remote PE connector.

Set up a UA Domain User

The Universal Adapter service must be able to access the relevant mailboxes hosted on each input and output Exchange server. Specifically, the Universal Adapter service must run as a domain user who can access these mailboxes. Therefore, create a new domain user for the Universal Adapter server or choose an existing domain user. Throughout this guide, the term 'UA domain user' refers to the domain user under which the UA service runs.

To configure the UA domain user

1. Create a new user or select an existing user to be your UA domain user.
2. Verify that the UA domain user belongs to the Local Administrators group on the Universal Adapter server.
3. Grant the UA domain user access to the relevant Exchange mailboxes:

   Exchange 2003

   Using Active Directory Users and Computers, go to the Exchange Advanced tab and click the Mailbox Rights button. Then, grant the UA domain user the Full mailbox access security privilege on the mailboxes that the Universal Adapter imports from and outputs to.

   Exchange 2007 and 2010

   To enable access to Exchange 2007 or 2010 mailboxes, use the Exchange Management Shell utility. For each mailbox that you want to access, run the following command:

   ```
   Add-MailPermission "<Mailbox>"
     -User "<Trusted User>"
     -AccessRights FullAccess
   ```

   Where:

   <Mailbox> is the display name of the mailbox the UA must access.

   <Trusted User> is the display name of the UA domain user.

Install the De-duplication Database

The de-duplication database is optional. If you do not install one the Universal Adapter will continue to process emails, but will be unable to remove duplicate emails.

Requirements

Note: If you do install the de-duplication database, you must do so before installing the Universal Adapter.

- The machine that will host the Universal Adapter de-duplication database must have Microsoft SQL Server 2005 or 2008 installed.
- Only one server hosts the de-duplication database. However, multiple Universal Adapters can access the same de-duplication database.
- In addition to selecting the De-Duplication Support feature, to use the de-duplication database you must also configure the DeDuplicate registry value for each mailbox input and follow the database installation instructions below.

Installation Steps

1. From the CA DataMinder distribution media, copy the contents of the \support\UA_DDUP folder to any location on the machine that will host the Universal Adapter de-duplication database.
2. To create the de-duplication database, tables and stored procedure, run the following command from the \support\UA_DDUP folder from a Windows command prompt:

   ```
   ua_dbms <user> <password>
   ```

   Where <user> and <password> parameters are the SQL Server de-duplication database user credentials. This will create the necessary tables.
3. Using the SQL Enterprise Manager, assign the DB_Owner privilege to the UA domain user for the UA_DDUP database. This privilege provides the UA domain user with sufficient access rights to the de-duplication database.

Install the Universal Adapter

If required, you can install multiple Universal Adapters on separate servers.

Note: If you want to install the de-duplication database, you must do so before installing the Universal Adapter.

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 Universal Adapter and then click Install.

   This launches the CA DataMinder Universal Adapter installation wizard in a separate window.

4. In the Universal Adapter installation wizard, navigate to the Customer Setup screen.

5. In the Custom Setup screen, choose the components that you want to install.

   CA Universal Adapter

   You must select this option.

   De-Duplication Support

   This feature is optional. It provides connectivity to the de-duplication database. If you do not select this option, the Universal Adapter will be unable to remove duplicate emails.

6. If you selected De-Duplication Support, enter details about your chosen database in the De-Duplication Database Location screen.

   Specify the name of the server hosting the database, or type 'localhost' to specify a database on the local machine.

   Note: If the installation wizard cannot validate the host server (for example, because it is not switched on), it adds a Bypass Validation check box to the screen. You can select this check box to skip the validation, but ensure you have correctly spelt the server name! if you have not, the Universal Adapter will be unable to subsequently connect to the de-duplication database.

7. In the Server Selection screen, select whether to use Microsoft Exchange or IBM Domino as the host server for your input mailboxes.

8. In the Universal Adapter Account screen, click the browse button to display the Service Credentials popup. Specify the user name and password for the Windows account that the Universal Adapter is to run under. This is the UA domain user you set up.

9. If you selected Microsoft Exchange in step 7, enter details about the Exchange Server machine name and mailbox name.

If you selected IBM Domino in step 7, enter the password for the Domino default user account. This account must be a valid Notes user.

10. Click Finish to complete the installation wizard.

Set Up Policy Engine Integration

You can configure the Universal Adapter to integrate with policy engines. To do this, you need to install and configure your policy engines and then install a Remote Policy Engine (PE) Connector. The Remote PE Connector functions as a policy engine hub to connect the UA to one or more policy engines.

Important! The Universal Adapter cannot integrate directly with policy engines, only via a policy engine hub.

Setting Up Integration

To enable integration between the Universal Adapter and policy engines

1. **Install one or more policy engines.**

   Run the CA DataMinder Server installation wizard—see the Platform Deployment Guide; search for 'policy engines, deployment'.

2. **Install and configure a policy engine hub.**

   Run the Integration Agents installation wizard to install the Remote PE Connector.

3. **Provide PE domain user credentials.**

   You must provide the policy engine hub service with the PE domain user credentials—see step 4 opposite.

4. **Configure the Universal Adapter.**

   To enable the connection between the Universal Adapter and the policy engine hub, ensure that the ApplyPolicy registry setting to Yes for each input mailbox.

5. **Configure your outputs**

   Set the OutputSmartTags registry setting to Yes for the outputs you want to support smart tags by.

6. Set up your policy engines with the required policies and smart tags.

   For full policy engine details, see the Policy engine chapter in the *Platform Deployment Guide*; search for 'policy engines'. For smart tag details, see the *Administrator Guide*; search for 'Smart Tagging'.

More information:

Set Up Policy Engine Integration (see page 246)
Set Up the Input Mailbox Subkeys (see page 259)

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 Remote PE Connector

The Universal Adapter can only integrate with policy engines via the Remote PE Connector. This is a subfeature of the External Agent. To install the Remote PE Connector, you must install the External Agent using the CA DataMinder Integration Agents installation wizard.

1. To launch the Integration Agents installation wizard, run setup.exe. Find this in the \Integration folder on your CA DataMinder distribution media.
2. In the Custom Setup screen, choose the External Agent API feature and click Next.
3. In the External Agent API configuration screen, ensure the Install Remote Policy Engine Connector check box is selected.
4. In the Policy Engine Hub Configuration screen, provide the credentials for the PE domain user (as specified in the *Platform Deployment Guide*. Search for 'PE domain user').

   Note: If you subsequently need to change these credentials, see the *Platform Deployment Guide*. Search for 'PE domain user, policy engine hub service'.
5. The installation wizard now has all the information it needs. Click Install to start the file transfer.

Configuring the Universal Adapter

Configuring the Universal Adapter is a multi-step process:

1. Configure the general operational registry settings in the \Universal Adapter registry key.
2. Configure the de-duplication database, using the \DeDuplicationDatabase registry subkey settings.

   Note: This step is optional.
3. Set up and configure the input source structure.
4. Set up and configure the output structure.
5. Set up the Unique ID property list for Exchange or Domino.
6. Create your LDAP connection for Exchange or Domino.
7. Set up integration with policy engines.

   Note: This step is optional.

Universal Adapter Registry Values

The table below describes the available registry values for the Universal Adapter. To configure the Universal Adapter, you can find these registry values in the following registry key:

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

Registry values are listed in alphabetical order. Registry keys are based on the example registry subkey diagrams, such as the one in Configure the General Operational Registry Settings.

The Registry values are as follows:

Universal Adapter key

UAMailboxNameExch*

UAMailboxServerExch*

UpdateConfig

FailedRetryIntervalMinutes

TotalRetryTimeMinutes

UALocalPSTFolder

WorkerThreadCount

LogFilePath

LogLevel

LogMaxNumFiles

LogMaxSizeBytes

DeDuplicationDatabase key

ServerName

ServerPort

ServerType

Templates key

Outputs

Type

DeDuplicate

DeEnvelope

ExpandDLs

FailedMailboxFolder

FailBlockedMessages

RetryFAILEDFolderMessages

ApplyPolicy

MessageClassIncludeFilter

MessageClassExcludeFilter

RecipientAddressIncludeFilter

RecipientAddressExcludeFilter

SenderAddressIncludeFilter

SubjectIncludeFilter

SubjectExcludeFilter

Mailbox subkeys

MailboxName

ServerName

Outputs

BaseTemplate

Enabled

EVF output pool <EVFPool1>

Type

EVF output location <EVFLocation1>

EVFPath

MinDiskSpaceMb

Exchange output pool <MailboxPool1>

OutputSmartTags

SecondaryOutputDatatype

SecondaryOutputs

StoreUniqueID

Type

Exchange output location <OutputMailbox1>

MailboxName

ServerName

Domino output pool <MailboxPool1>

OutputSmartTags

StoreUniqueID

Type

Domino output location <OutputMailbox1>

MailboxName

ServerName

DLL output pool <StorageSolution1>

SecondaryOutputs

StoreUniqueID

Type

DLL output location <ArchiveLocation1>

COMProgID

UniqueIDPropListExch key

EnvelopeInnerOrder

EnvelopeOuterOrder

HashOrder

UniqueIDPropListNotes key

HashOrder

More information:

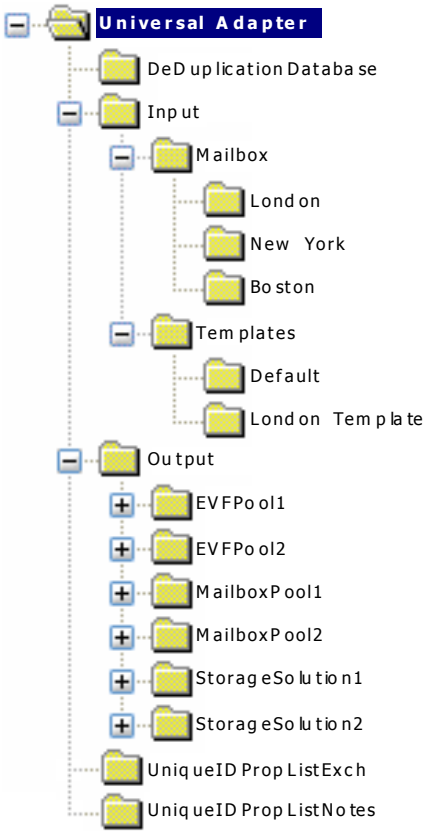
Configure the General Operational Registry Settings (see page 251)
MailboxName (see page 277)
ServerName (see page 277)

Configure the General Operational Registry Settings

The Universal Adapter registry key contains registry values to control the general operation of the Universal Adapter. All values are initialized to sensible defaults at installation, with two exceptions—see the warning below:

Important! UAMailboxNameExch and UAMailboxServerExch registry values are Exchange-only registry values and must be configured before you run the Universal Adapter service for the first time.

Example registry subkeys:



The registry values for the Universal Adapter registry key are listed as follows:

Universal Adapter key

UAMailboxNameExch *

UAMailboxServerExch *

UpdateConfig

FailedRetryIntervalMinutes

TotalRetryTimeMinutes

UALocalPSTFolder

WorkerThreadCount

LogFilePath

LogLevel

LogMaxNumFiles

LogMaxSizeBytes

* Only applicable to Exchange emails.

Universal Adapter Registry Key

To set up the general Universal Adapter registry settings, you need to configure the registry values in the following registry key, created during installation:

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

The registry values for the UniversalAdapter registry key are:

UpdateConfig

Type: REG_DWORD

Data: Enables administrators to update the registry while the Universal Adapter service is running. Set to 1 to force the Universal Adapter to re-read the registry. When the Universal Adapter has accepted the changes to the registry, it automatically resets this registry value to 0, so that this task is not carried out each time the registry is updated. If the Universal Adapter fails to accept the changes, it is automatically set to 2.

UAMailboxNameExch

Type: REG_SZ

Data: This Exchange-only value specifies the name of the Exchange mailbox for the UA domain user. That is, the user under which the Universal Adapter service runs.

Note: You must set UpdateConfig to 1 for changes to this registry setting to take effect.

UAMailboxServerExch

Type: REG_SZ

Data: This Exchange-only value specifies the name or IP address of the Exchange server hosting the mailbox for the UA domain user.

Note: You must set UpdateConfig to 1 for changes to this registry setting to take effect.

FailedRetryIntervalMinutes

Type: REG_DWORD

Data: Specifies how long (in minutes) the Universal Adapter waits between attempts to re process a failed e-mail. If the time spent trying to re-process the email reaches that specified by TotalRetryTimeMinutes, the e-mail is moved to the \Failed folder. The location of the \Failed folder is specified in the FailedMailboxFolder registry subkey and is specifically for each input source.

TotalRetryTimeMinutes

Type: REG_DWORD

Data: Specifies how long (in minutes) the Universal Adapter will try to re-process a failed e-mail. This is set to 1440 (24 hours) by default, to prevent the \Failed folder from becoming populated too quickly in the event of an output access problem.

UALocalPSTFolder

Type: REG_ SZ

Data: Specifies the location of the .PST file used to temporarily store emails while the Universal Adapter processes them. If this setting is left blank, the .PST file location defaults to:

```
\Documents and Settings\All Users\Application Data\CA\CA DataMinder\data
```

Note: You must set UpdateConfig to 1 for changes to this registry setting to take effect.

WorkerThreadCount

Type: REG_DWORD

Data: Defaults to 10. Specifies the total number of concurrent worker threads used by the Universal Adapter to process e mails. This is the total number of worker threads across all inputs. The maximum number of worker threads is 100. Even if you specify more than 100 worker threads in this registry value, the count will remain 100.

LogFilePath

Type: REG_ SZ

Data: Defaults to the \UA\Log subfolder in the CA DataMinder installation folder on the Universal Adapter host machine. Specifies the folder you want to write log files to. Multiple log files are created, depending on the LogMaxNumFiles registry value.

Note: The UA domain user must have write access to the specified folder.

LogMaxNumFiles

Type: REG_DWORD

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

LogLevel

Type: REG_DWORD

Data: Defaults to 2. Determines the verbosity of logging for the Universal Adapter. For example, you can configure the Universal Adapter to only log errors or important system messages. Log entries are written to the UASrv_<date>.log file, where <date> is the date and time when the log file was created. The location of the log file is specified in the LogFilePath registry setting (see above).

The supported logging levels are:

0 No log entries are written, except for basic start and stop messages

1 Errors only

2 Errors and warnings

3 Errors, warnings, plus any extra information available

4 Errors, warnings, plus any extra information available and trace

Note: Setting LogLevel to 4 will cause the log file to grow extremely rapidly. This level of logging is provided for testing and diagnostic purposes. For example, it shows storage and retrieval on every resource item.

LogMaxSizeBytes

Type: REG_DWORD

Data: Defaults to 1,000,000 (1MB). Specifies the maximum size for each log file. When the current log file reaches its maximum size, the Universal Adapter creates a new log file. Log entries are written to a UASrv_<date>.log file.

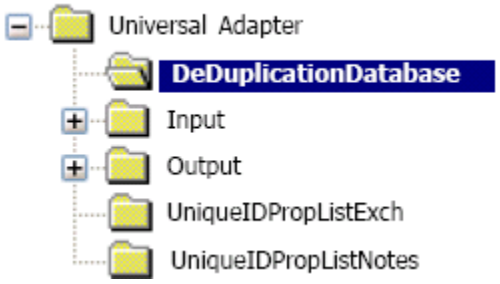
Configure the De-duplication Database

To set up the de-duplication database registry settings, you need to configure the registry values in the following registry key, created during installation:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
    \CurrentVersion\UniversalAdapter\DeDuplicationDatabase
```

You need to configure this subkey to specify the location and type of the server hosting the de-duplication database.

Example: Universal Adapter registry keys: DeDuplicationDatabase subkey



Note: If you are not using a de-duplication database, you do not need to configure settings within this registry key.

DeDuplicationDatabase Registry Key

This subkey contains the following registry values:

ServerName

Type: REG_SZ

Data: Specifies the name or IP address of the SQL server hosting the de-duplication database. This is automatically set up if you installed the De-Duplication Support feature. However, you still need to configure the DeDuplicate registry value for each mailbox input.

Note: You must set UpdateConfig to 1 for changes to this registry value to take effect.

ServerPort

Type: REG_ WORD

Data: Specifies the port number used to connect to the de-duplication database. We recommend that you do not change this registry value.

ServerType

Type: REG_ SZ

Data: Specifies the database engine for the de-duplication database. This **must** be set to MSSQL. Only Microsoft SQL Server is supported in the current release.

Set Up and Configure the Input Source Structure

The Universal Adapter can import emails from multiple journal mailboxes. Each input source mailbox must:

- Have its own registry subkey, even if the mailboxes are all hosted on the same Exchange server.
- Reference a parent template (even if you only have one input source).

The procedure for setting up the input source structure is summarized below:

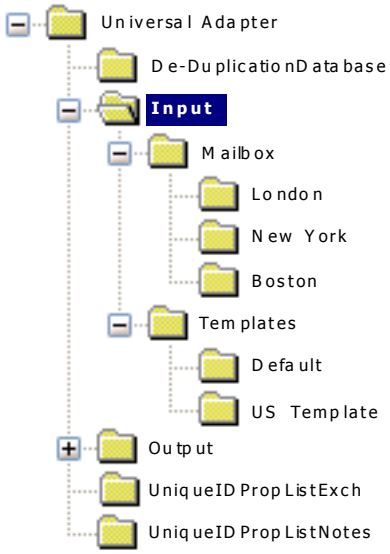
1. Set up input template subkeys
2. Set up input mailbox subkeys

To set up the input source structure, you need to configure Universal Adapter registry values in the following registry key, created during installation:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
    \CurrentVersion\UniversalAdapter\Input
```

Input Mailbox Overview

You need to create a mailbox subkey for each input mailbox you want to import emails from. For example, if you are importing from three journal mailboxes, each from a different office branch, you need to create three registry subkeys with meaningful names, such as London, New York and Boston:



Example Universal Adapter registry: Input subkey

In this example:

1. The London, New York and Boston input source mailboxes each have their own registry subkey.
2. There are two templates; the default template and a custom template.

Input Templates Overview

Each input source mailbox must have its own registry subkey and parent template.

A parent template is a set of registry values contained in a custom subkey within the Templates registry key. Using a parent template enables you to set the value of registry settings for multiple 'child' mailboxes. For this reason, we recommend that a template contains only those registry values that are common to all its child mailboxes; if you need to make custom changes to an individual mailbox source, you must configure the registry settings within that mailbox source.

To reference a parent template to a child mailbox, configure the BaseTemplate registry value within the mailbox subkey.

Notes:

- You can create multiple templates, but only one can be used per mailbox.
- If a registry value is set within a mailbox subkey *and* a template, the Universal Adapter uses the value in the mailbox subkey.
- If any value in the input is left blank, the Universal Adapter looks for that registry value first in the template, and if not there, uses CA DataMinder hard coded defaults. We recommend that you do not rely on these defaults, as they could potentially change in a future version.
- Some values (for example, the filters) do not have defaults. If these values are not specified by either the input or its template, then an invalid configuration is reported and the input is disabled.

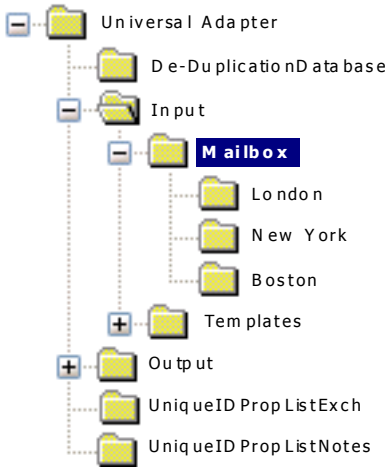
Set Up the Input Mailbox Subkeys

You need to create individual subkeys for each input mailbox in the Mailbox subkey.

To set up mailbox registry settings, you need to configure the registry values in the following registry key, created during installation:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
  \CurrentVersion\UniversalAdapter\Input\Mailbox
```

You need to configure this subkey to specify



Each input mailbox can only have one parent template. There are certain registry values that are not appropriate for use with a template. We recommend that these values (listed opposite) are configured within the input mailbox subkey.

You can also configure registry settings within an input mailbox subkey to overwrite values in the template.

In each of the registry subkeys given in the example above (London, New York and Boston), you need to specify the settings that are unique to that mailbox and not listed in the template.

The following registry values are mandatory or recommended for the mailbox subkey:

Mailbox subkey

MailboxName

ServerName

Outputs

BaseTemplate

Enabled

More information:

Input Templates Overview (see page 258)
Set Up the Input Template Subkeys (see page 266)

Templates Subkey

This subkey contains the following registry values:

Outputs

Type: REG_SZ

Data: Specifies a comma-separated list of the outputs that the Universal Adapter will output e-mails to. For example, MailboxPool1 and MailboxPool2. This list must match the subkey names listed under the Output registry key.

Type

Type: REG_SZ

Data: Set this to Exch or Notes and specifies whether the mailbox uses Microsoft Exchange or Domino Server. All mailboxes must use the same type.

DeDuplicate

Type: REG_DWORD

Data: Defaults to 0. Set this to 1 or 0. If set to 1, the Universal Adapter will process e-mails via the de-duplication database. If set to 0, de-duplication is disabled.

Notes:

- You must have previously installed the De-Duplication Support option during the Universal Adapter installation.
- This value applies to Universal Adapter outputs only.

DeEnvelope

Type: REG_DWORD

Data: Defaults to 1. This Exchange-only value can be set to 1 or 0 and specifies whether de-enveloping is enabled. If set to 1, any e-mails being imported that had previously been through the Microsoft Exchange envelope journaling process are de-enveloped.

Any emails in the mailbox that have not been through this process are simply imported in their current format. That is, they are not de-enveloped.

Note: This value applies to mailbox outputs only.

DuplicatesDatabase

Type: REG_SZ

Data: Specifies the name of the Domiono database used to store duplicate emails.

Note: This setting is provided for testing and diagnostic purposes.

Note: This setting is only applicable when processing emails in a Domino mailbox. If processing Exchange emails, use the DuplicatesMailboxFolder registry value.

DuplicatesMailboxFolder

Type: REG_SZ

Data: Specifies the name of the folder within the Exchange mailbox used to store duplicate emails. When identifying a folder, use backslashes as path separators. For example, \Duplicates will place all duplicate emails in a subfolder named Duplicates in the root of the mailbox. If this registry setting does not exist and de-duplication is enabled, then duplicate emails will simply be deleted.

Note: This setting is provided for testing and diagnostic purposes.

Note: This setting is only applicable when processing emails in an Exchange mailbox. If processing Domino emails, use the DuplicatesDatabase registry value.

ExpandDLs

Type: REG_SZ

Data: Defaults to No. Applies to Universal Adapter outputs only. Set this to Yes or No to specify whether the UA will expand an e-mail's distribution lists before it is output. If set to Yes, you also need to specify a threshold number of list members. For example, to expand up to 20 members of a distribution list, set this registry value to Yes, 20.

Notes:

- If the number of entries in the distribution list exceeds this value, then the distribution list is not expanded at all.
- If ApplyPolicy is set to Yes, then distribution lists are always expanded as emails are processed by the policy engine. If ApplyPolicy is set to Yes, then we recommend you set ExpandDLs to 0 to avoid any unnecessary processing.

After expanding this number of recipients from the distribution list, or if the Universal Adapter detects that expanding a nested distribution list would exceed this number, no further individual recipients are extracted for that DL. Other remaining DLs may still be expanded.

Note: This ExpandDLs setting is ignored for envelope journaled emails as DLs have already been expanded by the Exchange servers.

FailedMailboxFolder

Type: REG_SZ

Data: Specifies the name of the folder within the mailbox used to store emails that the UA could not process. For example, this can happen if the content of the email is corrupted, or because the server hosting the output is unavailable. When identifying a folder, use backslashes as path separators. For example, \Failed will place all failed emails in a subfolder named Failed in the root of the mailbox.

FailBlockedMessages

Type: REG_DWORD

Data: Defaults to 0. If set to 1, moves any currently failing e-mails directly to the \Failed folder, and does not try to reprocess them. This registry value returns to its default setting of 0 after any currently failing emails are moved to the \Failed folder.

Important! Setting this registry value has no affect on future failing emails. It only moves currently failing emails.

RetryFailedFolderMessages

Type: REG_DWORD

Data: Moves failed emails back to the Inbox so they can be re-processed.

ApplyPolicy

Type: REG_MULTI_SZ

Data: Defaults to No. This value can be set to Yes or No and specifies whether the UA connects to a policy engine (via a hub) to apply policy and generate smart tags for the emails being processed.

Note: The UA can only connect to a policy engine hub; it cannot connect directly to a policy engine.

Include and Exclude Filters

For each pair of Include and Exclude filters, the Include filter is processed first. This means that if an e-mail does not match the value of the Include filter, the Universal Adapter infers that the email has been successfully processed and therefore deletes it. An email is only processed by the Exclude filter if it matches the value of the Include filter.

If an email does match the Include filter, it is then checked against the Exclude filter. If this matches, the e-mail is deemed successfully processed, and therefore deleted. So, for an email to be processed by the Universal Adapter, it must match all Include filters, and must not match any Exclude filters.

MessageClassIncludeFilter, MessageClassExcludeFilter

Type: REG_MULTI_SZ

Data: Enable the Universal Adapter to only process, or exclude from processing, emails of specific categories. For example:

To **only** process appointment notification emails, set the MessageClassIncludeFilter value to:
`IPM.Taskrequest`

To import **all** categories of email, set this value to * (an empty value fail).

To exclude **all** faxes and meeting requests, set this value to:
`IPM.*.Fax, IPM.Schedule.Meeting.Request`

Note: You can specify multiple inclusions or exclusions using a comma-separated.

RecipientAddressIncludeFilter, RecipientAddressExcludeFilter

Type: REG_MULTI_SZ

Data: Enable the Universal Adapter to only process or exclude from processing, e-mails sent to specific users. Recipient addresses that match the address specified in:

- RecipientAddressIncludeFilter are the only addresses to be processed.

  To process emails sent to any recipient, ensure this registry value is set to * (an empty value will fail).

- RecipientAddressExcludeFilter are not processed.

Note: You can specify multiple addresses in either value using a comma-separated list.

When setting either of these filters, be aware that the Universal Adapter only retrieves one email address for each recipient, as it does not perform an address book lookup. For internal recipients, this will normally be the Exchange email address, but for external recipients, it can be the Exchange or SMTP email address.

The syntax of this setting is the same as the CA DataMinder email address filter syntax. For details, see the Administration console online Help; search the index for ‘wildcards’.

For assistance, contact CA Support at http://ca.com/support.

SenderAddressIncludeFilter, SenderAddressExcludeFilter

Type: REG_MULTI_SZ

Data: Enables the Universal Adapter to only process or exclude from processing, e-mails sent by specific users. Sender addresses that match the address specified in:

- SenderAddressIncludeFilter are the only addresses to be processed.

  To process emails sent to any recipient, ensure this registry value is set to * (an empty value will fail).

- SenderAddressExcludeFilter not imported.

Note: You can specify multiple addresses in either value using a comma-separated list.

When setting this filter, be aware that the Universal Adapter only retrieves one email address for each sender, as it does not perform an address book lookup. For internal recipients, this will normally be the Exchange email address, but for external senders it can be the Exchange or SMTP email address.

The syntax of this setting is the same as the CA DataMinder email address filter syntax. For details, see the Administration console online Help; search the index for 'wildcards'.

For assistance, contact CA Support at http://ca.com/support.

SubjectIncludeFilter, SubjectExcludeFilter

Type: REG_MULTI_SZ

Data: Enables the Universal Adapter to only import or exclude from import, emails with a specific subject. E-mails with the subject matching the one specified in:

- SubjectIncludeFilter are the only emails to be imported.

To import emails of all subjects, ensure this registry value is set to * (an empty value will fail).

- SubjectExcludeFilter are not imported.

Note: You can specify multiple subjects in either value using a comma-separated list.

When using any of the following special characters in their literal sense, you need to prefix them with a \ backslash character.

{ } | [ ] % ? * \

For example, to search for '24*7', you need to enter '24*7'.

If you need further guidance, see the 'Search text variables' technical note. This is available from CA Support: http://ca.com/support.

More information:

Filtering on Notes Journal Properties (see page 266)

Filtering on Notes Journal Properties

The Universal Adapter needs access to the information it filters on. If you are using a Notes Journal property in an Include or Exclude filter, you must ensure that it is excluded from encryption. To do this, add it to the Notes Journal 'Field encryption exclusion list'. For example:

To filter on	Exclude this property from encryption
Subject	Subject
Sender	From
Recipient	SendTo, CopyTo
Form	Memo

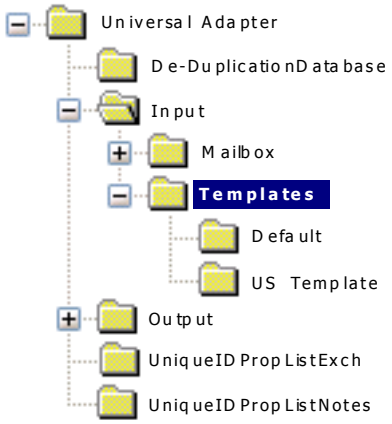
Set Up the Input Template Subkeys

You can create custom templates in the Templates subkey. You can create multiple templates, but an input mailbox can only have one parent template.

To set up template registry settings, you need to configure the registry values listed in the Templates registry key, created during installation:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
  \CurrentVersion\UniversalAdapter\Input\Templates
```

You need to configure this subkey to specify



Example Templates registry key

The Universal Adapter first checks to see if a registry value has been explicitly set within a mailbox subkey. If it has not, the Universal Adapter uses the corresponding value defined in the parent template of the mailbox. That is, the template specified in the mailbox BaseTemplate registry value.

To configure the template registry settings, specify the settings that are common to each mailbox and not included in the custom settings in the mailbox subkey. The recommended registry values for the Templates registry value are:

Templates subkey

Outputs

Type

DeDuplicate

DeEnvelope

DuplicatesMailboxFolder

ExpandDLs

FailedMailboxFolder

FailBlockedMessages

RetryFAILEDFolderMessages

ApplyPolicy

MessageClassIncludeFilter

MessageClassExcludeFilter

RecipientAddressIncludeFilter

RecipientAddressExcludeFilter

SenderAddressIncludeFilter

SubjectIncludeFilter

SubjectExcludeFilter

SubjectExcludeFilter

More information:

Input Templates Overview (see page 258)
Templates Subkey (see page 261)

Mailbox Subkey

The recommended registry values for the Mailbox registry value are:

MailboxName

Type: REG_SZ

Data: This mandatory value specifies the name of the Exchange or Domino mailbox you want to import from.

For Exchange

You must supply the full Exchange mailbox name unless the last CN= component is known to be unique. If this is so, you need only specify this last component value to identify the mailbox. For example, for an Exchange address of:

```
/O=UNIPRAXIS/OU=FIRST ADMINISTRATIVE GROUP
/CN=RECIPIENTS/CN=SRIMMEL
```

The value SRIMMEL is sufficient for the Exchange server to resolve the address and identify the mailbox.

Note: Supplying the value SRIMMEL is sufficient as long as this value does not form the first part of another CN= component, for example, SRIMMEL2 or SRIMMELL.

To import from more than one mailbox, each mailbox must have a separate subkey in the Input\Mailboxes registry key.

For Domino

You must supply the mailbox name. For example, mail\srimmel.

Note: If this mailbox is a 'mail-in' journal, then the Notes user that the UA uses to connect to Domino will be added as a participant of any e-mails processed by Domino.

ServerName

Type: REG_SZ

Data: This mandatory value specifies the name or IP address of the Exchange or Domino server hosting the mailbox you want to import from.

Outputs

Type: REG_SZ

Data: This mandatory value is a comma-separated list of the outputs that the Universal Adapter will output emails to. For example, MailboxPool1 and MailboxPool2. This list must match the subkey names listed under the Output registry key.

BaseTemplate

Type: REG_SZ

Data: Specifies which template the mailbox settings are taken from.

Enabled

Type: REG_DWORD

Data: This value can be set to 1 or 0 and specifies whether the Universal Adapter will process the mailbox. If set to 1, the Universal Adapter will process the mailbox. For example, you may want to temporarily disable importing from a specific mailbox in order to carry out maintenance work.

Set Up the Output Structure

The Universal Adapter can output emails to any of the following output types:

- Microsoft Exchange mailbox
- Domino Server mailbox
- EVF files
- Third Party DLLs

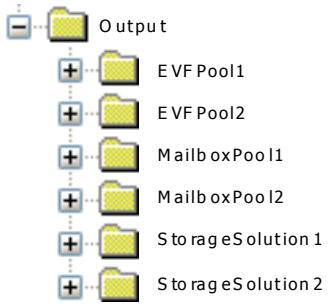
The procedure for setting up the output structure is summarized below:

1. Configure the Output registry key.

   In the **Output** registry key, create a list of 'pool' subkeys: one for each output you want to send e-mails to. The subkey names must match those specified in the Outputs registry value in the Input registry key. For example, if the Outputs values for each output type are:

   ```
   EVFPool1, EVFPool2
   MailboxPool1, MailboxPool2
   StorageSolution1, StorageSolution2
   ```

   Then the output structure would look like this:



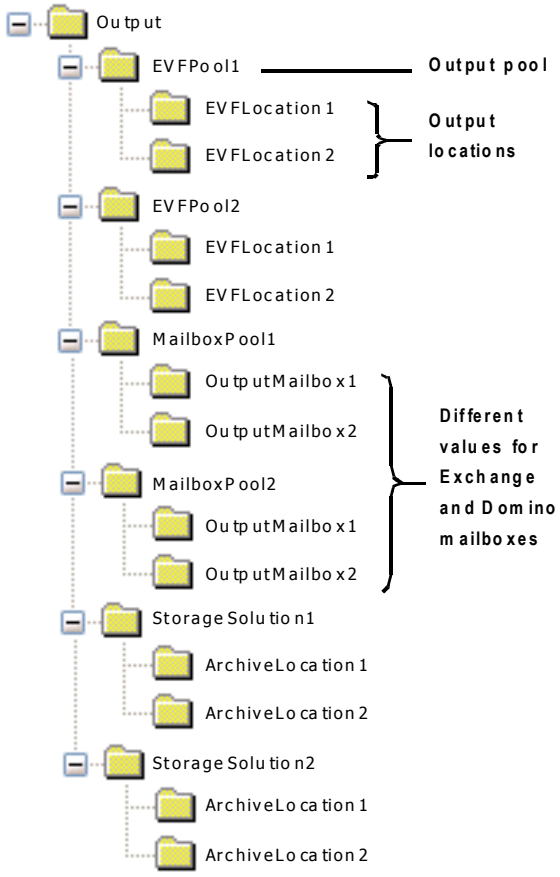
2. Configure output pool subkeys.

   These registry subkeys, created in step 1, represent groups (or pools) of each output type. Such output pooling provides optimum load-balancing. Within each output pool subkey, you must set up further location subkeys to define output pool structure. To provide maximum throughput, the Universal Adapter processes emails to outputs in round robin fashion.

   For example, if you import emails from two separate mailboxes, it is more efficient to output them to at least two mailboxes, rather than just one. When processing e-mails to **MailboxPool1**, the Universal Adapter outputs an email first to **OutputMailbox1**, then **OutputMailbox2**, then again to **OutputMailbox1**.

3. Configure output location subkeys.

   For each output pool, you then need to create location subkeys for each journal mailbox, EVF file or third party archive location that you want the UA to output emails to. For example:



   Even if you only want to output to one mailbox or one EVF file location, you still need to add its registry subkey to a mailbox or EVF pool. For example, **OutputMailbox1** or **EVFLocation1**.

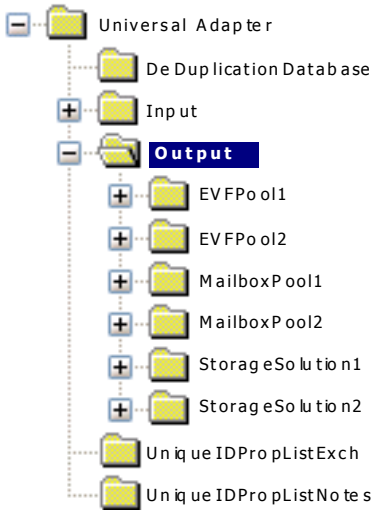
Output Registry Key

To set up the output structure, you need to configure the Universal Adapter registry settings on the UA host server by locating the following registry key, created during installation:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
  \CurrentVersion\UniversalAdapter\Output
```

Example registry keys

An example Output key and subkeys are shown below:



The following sections describe the registry values for each of the three output formats.

EVF Pool Subkeys

For each EVF output pool, you must create a location subkey. These location subkeys must be grouped in pools and emails are then output to EVF files in any of the location subkeys in the specified output pool.

The required registry value for the pool subkeys is:

Type

Type: REG_SZ

Data: This must be set to EVF to output emails to individual .EVF files.

More information:

Individual EVF Location Subkeys (see page 272)

Type

Type: REG_SZ

Data: This must be set to EVF to output emails to individual .EVF files.

Individual EVF Location Subkeys

Each location subkey must contain the following registry values: EVFPath and MinDiskSpaceMb. In the example in Output Registry Key, this applies to EVFLocation1 and EVFLocation2 subkeys.

Type

REG_SZ

Data

Specifies the path and folder to where you want to output the .EVF files.

Type

REG_DWORD

Data

Defaults to 100. This specifies the minimum amount of free disk space (in Mb) required on the drive containing the EVF output folder. If the free space drops below this threshold, the Universal Adapter will stop processing input mailboxes using this output until the free space recovers above the threshold.

Journal Mailbox Pool Subkeys

More information:

Exchange Journal Mailbox Pool Subkeys (see page 273)
Domino Journal Mailbox Pool Subkeys (see page 276)

Exchange Journal Mailbox Pool Subkeys

For each Exchange mailbox you want the UA to output e-mails to, you must specify an Exchange journal mailbox and the Exchange server hosting that mailbox. The location subkey must contain MailboxName and ServerName registry values: In the example on Set Up the Output Structure, this applies to the subkeys **MailboxPool1** and **MailboxPool2**.

The required registry values for this registry key are ilsted below.

Required registry values:

StoreUniqueID (see page 273)
Type (see page 273)
OutputSmartTags (see page 274)
SecondaryOutputs (see page 274)
SecondaryOutputDatatype (see page 274)
Individual Exchange mailbox location subkeys (see page 274)
MailboxName (see page 275)
ServerName (see page 275)

StoreUniqueID

Type

REG_SZ

Data

Defaults to No. Each email processed by the Universal Adapter can be given a unique ID. If set to Yes, this ID is stored as a specific MAPI property in the output email. You also need to specify here which MAPI property you want to use to store the ID. For example:

Yes,PS_MIME_HEADERS,NAME=X-Orch-UniqueID

The example above will store the unique ID in a MIME transmittable property named 'X-Orch-UniqueID'.

More information:

Configure the Unique ID Property List (see page 280)

Type

Type

REG_SZ

Data

This value must be set to EXCH to output e-mails to an Exchange mailbox.

OutputSmartTags

Type

REG_SZ

Data

Defaults to No. This value can be set to Yes or No and specifies whether each e-mail processed by the Universal Adapter can be populated with smart tags. The smart tag is then written to a PS Public strings property:

ORCH-XmlSmartTags

SecondaryOutputs

Type

REG_SZ

Data

Specifies the name of the secondary output pool. It must be set to the name of an existing EVF output pool. For example, EVFPool2 on Output Registry Key.

SecondaryOutputDatatype

Type

REG_DWORD

Data

Specifies the archive identifier, That is, the data type stored within the EVF file created as the secondary output. Together with the event identifier, it enables CA DataMinder's Remote Data Manager to retrieve events from a third party archive during a subsequent event search.

For a list of supported values, please contact CA Technical Support: http://ca.com/support.

Individual Exchange mailbox location subkeys

Each location subkey must contain the following registry values: MailboxName and ServerName. In the example on Set Up the Output Structure, this applies to **OutputMailbox1** and **OutputMailbox2** subkeys.

More information:

Set Up the Output Structure (see page 269)

MailboxName

Type

REG_SZ

Data

Specifies the name of the Exchange mailbox you want to output the processed emails to. You must supply the full Exchange mailbox name unless the last CN= component is known to be unique. If this is so, you need only specify this last component value to identify the mailbox. For example, for an Exchange address of:

```
/O=UNIPRAXIS/OU=FIRST ADMINISTRATIVE GROUP
/CN=RECIPIENTS/CN=SRIMMEL
```

The value SRIMMEL is sufficient for the Exchange server to resolve the address and identify the mailbox.

To output to more than one mailbox, each mailbox must have a separate registry key in the Output\Mailboxes registry key.

ServerName

Type

REG_SZ

Data

Specifies the name of the Exchange server hosting the mailbox you want to output e-mails to. For example, if the server has a name of MailboxSRV1, the ServerName value will be in a similar format to:

```
/O=UNIPRAXIS/OU=FIRST ADMINISTRATIVE GROUP
/CN=CONFIGURATION/CN=SERVERS
/CN=MAILBOXSRV1
/CN=MICROSOFT PRIVATE MDB
```

You must supply the full Exchange server name unless the *penultimate* CN= component is known to be unique. If this is so, you need only specify this *penultimate* component value to identify the server. For example, for the Exchange address above, the value MailboxSRV1 may be sufficient for the Exchange server to identify the server.

Note: Do not use the DNS name of the server. For example, MailboxSRV1.unipraxis.com.

Domino Journal Mailbox Pool Subkeys

For each Notes mailbox you want the UA to output e-mails to, you must specify a Domino journal mailbox and the Domino server hosting the mailbox. Domino journal mailboxes are encrypted for a particular user. In order for the Universal Adapter to access such mailboxes:

- The Universal Adapter must be running as the user the mailbox is encrypted for, OR
- The properties used to create the UniqueID must be excluded from encryption.

The location subkey must contain the following registry values: MailboxName and ServerName. In the example on Output Registry Key, this applies to **MailboxPool1** and **MailboxPool2** subkeys.

The required registry values for this registry key are ilsted below.

Required registry values:

StoreUniqueID (see page 276)
Type (see page 276)
OutputSmartTags (see page 277)
Individual Domino mailbox location subkeys (see page 277)
MailboxName (see page 277)
ServerName (see page 277)

StoreUniqueID

Type

REG_SZ

Data

Defaults to No. Each email processed by the Universal Adapter can be given a unique ID. If set to Yes, this ID is stored as a property in the output e-mail. You also need to specify here which property you want to use to store the ID. For example:

```
Yes, OrchUniqueID
```

The example above will store the unique ID in a property named 'OrchUniqueID'.

Type

Type

REG_SZ

Data

This value must be set to Notes to output e-mails to a Domino mailbox.

OutputSmartTags

Type

REG_SZ

Data

Defaults to No. This value can be set to Yes or No and specifies whether each e-mail processed by the Universal Adapter can be populated with smart tags. It is written to a Notes named property:

`ORCH-XmlSmartTags`

Individual Domino mailbox location subkeys

Each location subkey must contain the following registry values: MailboxName and ServerName. In the example in Set Up the Output Structure, this applies to **OutputMailbox1** and **OutputMailbox2** subkeys.

More information:

Set Up the Output Structure (see page 269)

MailboxName

Type

REG_SZ

Data

Specifies the name of the Domino mailbox you want to output the processed emails to. You must supply the location of the mailbox on the Domino Server. For example:

`mail/journal.nsf`

To output to more than one mailbox, each mailbox must have a separate registry key in the Output\Mailboxes registry key.

ServerName

Type

REG_SZ

Data

Specifies this is the name of the Domino server hosting the mailbox you want to output emails to.

Third party DLL Registry Subkeys

For each third party archive, you must create a registry subkey. These location subkeys must be grouped and emails are then processed to any of the location subkeys in the specified output location group. In the Output Registry Key example, this applies to **StorageSolution1** and **StorageSolution2** subkeys:

More information:

StoreUniqueID (see page 278)
Type (see page 278)
SecondaryOutputs (see page 279)
Archive Location Subkeys (see page 279)
COMProgID (see page 279)

StoreUniqueID

Type

REG_SZ

Data

Defaults to No. Each email processed by the Universal Adapter can be given a unique ID. If set to Yes, this ID is stored as a specific MAPI property in the output email. You also need to specify here which MAPI property you want to use to store the ID.

Type

Type

REG_SZ

Data

This value must be set to DLL to output e-mails to a third party archive system.

SecondaryOutputs

Type

REG_SZ

Data

Enables the Universal Adapter to create an EVF file after outputting to a third party DLL or Exchange mailbox. This enables you to import the EVFs into the CMS in order to integrate CA DataMinder with a third party archive. Type in the name of the EVF output pool you want the Universal Adapter to write the e-mail to. This must be a single EVF output name, for example:

```
IRM_EVF_Pool
```

Note that the resulting EVF files will contain event metadata provided to the Universal Adapter by the third party DLL. Specifically, this will include anT event identifier and an archive identifier. These identifiers enable CA DataMinder's Remote Data Manager to retrieve events from the third party archive during a subsequent event search.

Archive Location Subkeys

Each location subkey must contain the following registry value. In the example on Output Registry Key, this applies to **ArchiveLocation1** and **ArchiveLocation2** subkeys:

COMProgID

Type

REG_SZ

Data

Specifies a COM class name or GUID that contains an interface to a third-party archive system. The specific value of the COM class name or GUID depends on the archive system.

Configure the Unique ID Property List

An email's unique ID is generated from a set of MAPI properties from the email. These properties are encrypted using the SHA-256 algorithm. Each MAPI property used to generate the unique ID must be defined in the registry. A set of these will then be specified for each journal email type. That is, envelope-journaled e-mails are configured differently to non-envelope-journaled emails.

By default, the Universal Adapter installation sets the MAPI properties that we have found to be the most useful for generating the unique email IDs. Care must be taken to ensure that these properties can uniquely identify an e-mail and are identical for all duplicates of that email.

The procedure for setting up the unique ID property list is summarized below:

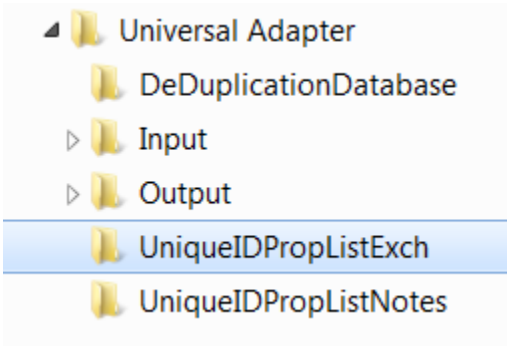
1. Configure the unique ID property list.
2. Create and configure custom registry values.

Note: If you change the MAPI properties used to create the unique ID (that is, you reconfigure any of the registry settings in the next section), the Universal Adapter could potentially allocate the same ID to two identical emails, causing duplicate emails to be processed and stored. For this reason, you must restart the UA service for the changes to take effect.

Configure the Unique ID Property List for Exchange

To set up the Universal Adapter so that it generates a unique ID for each email it processes, you need to configure the following registry key on the UA host server:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
  \CurrentVersion\UniversalAdapter\UniqueIDPropListExch
```



Example: Universal Adapter registry keys: UniqueIDPropListExch subkey

This key contains the registry values that are used to define how the MAPI properties are combined to create unique IDs.

Unique ID Registry Values

Note: You must set UpdateConfig to 1 for changes to the following registry settings to take effect.

HashOrder

Type: REG_SZ

Data: This value is only for emails that have **not** been through the envelope journaling process. It specifies a comma-separated list of MAPI properties used by the Universal Adapter to generate a unique ID for each email processed. For normal use, we recommend you use:

```
InternetMessageID, ClientSubmitTime
```

Some archive solutions (for example, ZANTAZ Digital Safe Adapter) replace the InternetMessageID property on each email. This makes the property unreliable for detecting duplicate messages amongst emails processed by other archive solutions.

In this situation, we recommend you use a combination of the following MAPI properties to generate the unique ID:

```
SubjectID, SenderNameID, DisplayToID, PlainTextBody, ClientSubmitTime
```

Note: Using the PlainTextBody property is an optional recommendation as it can potentially have an impact on processing time.

EnvelopeInnerOrder

Type: REG_SZ

Data: This value is only for emails that have been through the envelope journaling process. The Universal Adapter generates a unique ID for such an email using MAPI properties from both its inner part (the part that contains the original e-mail) and its outer part (the journal report containing the transport envelope data of the original email). The value specifies a comma-separated list of the MAPI properties for the inner part. For example:

```
ClientSubmitTime
```

EnvelopeOuterOrder

Type: REG_SZ

Data: This value is only for emails that have been through the envelope journaling process. The Universal Adapter generates a unique ID for such an email using MAPI properties from both its inner part (that is, the part that contains the original e-mail) and its outer part (that is, the journal report containing the transport envelope data of the original email). The value specifies a comma-separated list of the MAPI properties for the outer part. For example:

```
PlainTextBody
```

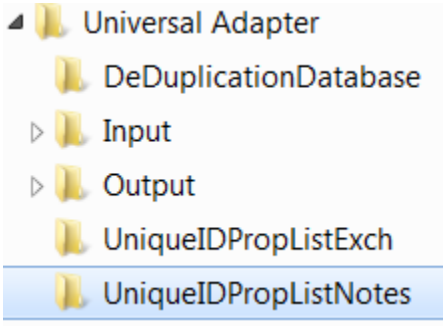
Definition of MAPI Properties

You define the MAPI properties listed in the registry values for HashOrder, EnvelopeInnerOrder, and EnvelopeOuterOrder, using registry values in the UniqueIDPropListExch key. For example, if HashOrder contains ClientSubmitTime, then a registry value UniqueIDPropListExch/ClientSubmitTime must define the value. Refer to "Create Property Definitions Registry Values for Exchange or Domino" for details.

Configure the Unique ID Property List for Domino

To set up the Universal Adapter so that it generates a unique ID for each Domino email it processes, you need to configure the following registry key on the UA host server:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA DataMinder
  \CurrentVersion\UniversalAdapter\UniqueIDPropListNotes
```



Example UA registry keys: UniqueIDPropListNotes subkey

This key contains the following default registry values, which are used to define how the properties are combined to create unique IDs.

Unique ID Registry Values

HashOrder

Type: REG_SZ

Data: This registry value specifies a comma-separated list of Notes properties (items or fields) that the Universal Adapter uses to generate a unique ID for each email processed.Use whatever properties (items or fields) that you have used elsewhere. We recommend you use:

```
MessageID, PostedDate
```

Note: You must set UpdateConfig to 1 for changes to this registry setting to take effect.

Definition of Notes Properties

Define the Notes properties listed in the HashOrder registry value using registry values in the UniqueIDPropListNotes key. For example, if HashOrder contains MessageID, then a registry value UniqueIDPropListNotes/MessageID must define the value. Refer to "Create Property Definitions Registry Values for Exchange or Domino" for details.

Create Property Definition Registry Values for Exchange (MAPI) or Domino (Notes)

You now need to add registry values for each property to be used to generate the unique ID. These registry values can be renamed if required.

Exchange Properties

To use the properties for Internet Message ID and Client Submit Time, you need to create REG_SZ registry values of InternetMessageID and ClientSubmitTime respectively in the UniqueIDPropListExch registry key and configure them to specify those properties. For example:

InternetMessageID

Type: REG_SZ

Data: COMMON, ID=0X1035001E

ClientSubmitTime

Type: REG_SZ

Data: COMMON, ID=0X00390040

Define MAPI Properties

MAPI properties exist in groups and either have an ID or a name specific to that group. This section defines the three types of MAPI property used most frequently.

Note: For any other MAPI groups, the GUID must be specified along with either the property ID or the name.

COMMON

These are the built-in MAPI properties. These properties must be referenced in the following format:

```
<COMMON>, <ID=0xvalue>
```

For example, to specify the MAPI property Internet Message ID, use the following:

```
COMMON, ID=0x1035001E
```

PS_PUBLIC_STRINGS {00020329-0000-0000-C000-000000000046}

This property must be referenced in the following format:

```
<PS_PUBLIC_STRINGS>,
<ID=0xvalue or NAME=value>
```

You must provide the MAPI group, followed by either the property ID or the property name. For example:

```
PS_PUBLIC_STRINGS, NAME=CustomProperty1
```

PS_MIME_HEADERS {00020386-0000-0000-C000-000000000046}

This property must be referenced in the following format:

```
<PS_MIME_HEADERS>, <ID=0xvalue or NAME=value>
```

You must provide the MAPI group, followed by either the property ID or the property name. For example:

```
PS_MIME_HEADERS, NAME=X-Archive-ID
```

The example will use the value set by the Internet-transmittable MIME tag X-Archive-ID.

Domino Properties

To use the properties for Message ID and Posted Data, you need to create REG_SZ registry values of MessageID and PostedData respectively in the UniqueIDPropListNotes registry key and configure them to specify those properties. For example:

MessageID

Type: REG_SZ

Data: $MessageID

PostedDate

Type: REG_SZ

Data: PostedDate

Define a Notes Property

Notes properties are simply named properties. To define a Notes property, you reference its name (but see the note in MessageID below).

Example:

- To specify the Notes property PostedDate, use:

  ```
  PostedDate
  ```

- To specify the Notes property MessageID, use:

  ```
  $MessageID
  ```

  Note: To specify the MessageID property, you need to prefix the value with a $ symbol.

Configuring Your LDAP Connection

You can configure a specific list of Global Catalog or LDAP servers for the Universal Adapter to use when expanding DLs. The procedure for configuring your LDAP connection is summarized below:

1. Set up the unique ID.
2. Configure the unique ID property list.
3. Create and configure custom registry values.

Windows 2003 Domains

By default, if the LookupLDAPServers registry value is not set, the Universal Adapter will use any available Global Catalog Server to expand DLs. No further configuration is necessary.

Configure a List of LDAP Servers

If importing from Domino, you must provide the list of LDAP servers.

Note: If importing from Exchange Server 2000, you do not need to configure your LDAP connection as the Universal Adapter automatically detects Active Directory.

To configure a list of LDAP servers

1. Locate the following registry key on the Universal Adapter host machine:

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

2. Within this registry key, you need to edit the following values:

LookupLDAPServers

Type

REG_MULTI_SZ

Data

Specifies the list of LDAP connection strings that will be used for DL expansion. The exact contents of an LDAP connection string will depend on the type of LDAP server.

Domino example: When connecting to a Domino server, the username must be the distinguished name of a user with read access to the relevant parts of the directory:
`cn=Spencer Rimmel:paris04@unipraxis`

Where:

- Spencer Rimmel is the username.
- paris04 is the password.
- unipraxis is the server name.

Although anonymous access (not supplying a username and/or password) may allow some attributes to be accessed, other attributes may be restricted and only accessible if you provide credentials.

LookupSearchFilter

Type

REG_SZ

Data

Used to exclude certain objects from a search. If you override the default filter, ensure that the new filter conforms to RFC 2254.

With Domino LDAP servers, both hidden and deleted members are returned when distribution list lookup operations use cn=admin mode. For example, to exclude deleted users, set this value to:

```
!(Is-deleted=true)
```

to ensure that deleted list members are filtered out.

LookupDirectoryType

Type

REG_SZ

Data

Specifies the type of lookup server. Set this value to LDAP to specify a set of LDAP servers. Otherwise, any available Global Catalog Server will be used.

With Domino LDAP servers, this registry entry must be set to LDAP and this also sets the authentication mode to be 'simple'. Simple authentication is required for the cn=admin indicator to be recognized.

LookupDirectoryBase

Type

REG_SZ

Data

Specifies the LDAP server's base DN or domain. This value defaults to empty. However, you can set this value to a specific base DN, for example, to speed up lookup operations or because the account used to access the LDAP directory only has permission to search a specific subset of the directory.

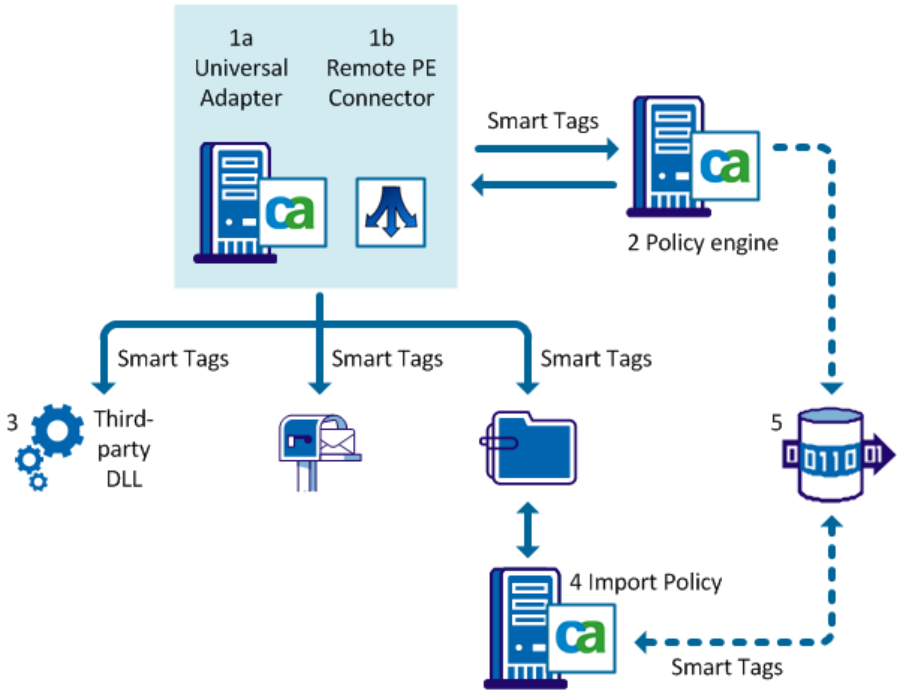
For Domino, set this registry value to the root domain of your organization, for example:

```
"o=unipraxis"
```

Policy Engine and Smart Tagging Integration

You can optionally link the Universal Adapter to a policy engine (via a hub), in order to apply policy and add smart tags to emails as appropriate. These emails are analyzed against policy as normal, optionally saved to the CMS and the smart tags added and passed back to the Universal Adapter.

Example: Universal Adapter: Smart Tagging using policy engines



1. **Universal Adapter.** You can optionally connect the Universal Adapter (**1a**) to a policy engine (**2**), via a hub (**1b**).
2. **Policy engine.** The policy engine applies policy to e-mails and populates them with smart tags before sending them back to the Universal Adapter (**1a**). It can also optionally save the processed emails as events, to the CMS (**5**).
3. **Outputs.** The Universal Adapter then outputs the e-mails, with the smart tag information, to: an Exchange or Domino mailbox; EVF files; or (for Exchange e-mails only) a third party DLL.
4. **Import Policy.** You can optionally use Import Policy to apply policy to the EVF files, and to generate smart tags, and replicate these up to the CMS (**5**).

More information:

Saving Events to the CMS (see page 289)
Adding Smart Tags to Emails (see page 289)

Saving Events to the CMS

We recommend that you configure your policy engine to save events to the CMS at the same time that it applies policy and adds smart tags to them. (For details, see the CA DataMinder Administration console online help; search the index for: 'captured data, emails'). This removes the need for Import Policy and therefore any additional policy engines. This is the most efficient process, unless your policy is particularly intense.

If this is the case (for example, your policy involves a deeper analysis of events), it may be more efficient to de-couple policy analysis from the processing of the Universal Adapter. In other words, we recommend that you do **not** configure the policy engine to save blob files to the database after applying policy and adding smart tags, but instead, send the blob files back to the Universal Adapter which can then output them to EVF files. These EVF files can then be processed via Import Policy and replicated up to the CMS. This ensures that the Universal Adapter rate is kept as high as possible and does not impact the archiving process.

Adding Smart Tags to Emails

Smart tags are generated by the policy engine as XML and passed back to the Universal Adapter, which then adds the XML to the email in a specified property.

The Universal Adapter can then output the email to a third party archive solution where its smart tag details can be processed. See below for an example of the Smart Tag XML data:

```
<?xml version="1.0" encoding="UTF-16" ?>
<apm schema_version="1" xmlns="http://www.orchestria.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.orchestria.com xmleventattributes.xsd">
<policy>
 <state>
  <smart_tags>
   <smart_tag name="st name1">
    <value>st value 1</value>
   </smart_tag>
   <smart_tag name="st name2">
    <value>st value 2</value>
    <value>st value 3</value>
   </smart_tag>
  </smart_tags>
 </state>
</policy>
</apm>
```

More information:

Exchange Journal Mailbox Pool Subkeys (see page 273)
Domino Journal Mailbox Pool Subkeys (see page 276)

Monitoring the Universal Adapter

This section focuses on diagnostics, error recovery and troubleshooting for the Universal Adapter. There are various sources of diagnostic information when monitoring the Universal Adapter. These are:

- Log files
- Performance counters
- Diagnostic files
- Error messages

Log Files

The Universal Adapter writes log entries to the log file, UASrv.log. These detail progress as each email is processed. This log file is in the \UA\Log subfolder in the CA DataMinder installation folder on the Universal Adapter host machine (unless the LogFilePath registry value has been set).

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

Performance Counters

The Universal Adapter includes two performance objects that are useful when diagnosing problems. In the Performance applet, you can add a range of useful performance counters.

1. Use the Performance applet, accessible from Administrative Tools to display counters for the Universal Adapter inputs and outputs.
2. Use the Add Counters dialog to specify which counters and, where relevant, instances you want to view for each performance object.

Performance Objects

The Universal Adapter supports the following performance monitor objects:

CA DataMinder UA Input Mailboxes

Multiple instances available; capped at 100 (see note below). This contains counters relevant to input journal mailboxes.

Available counters show statistics such as: the number and rate at which emails are being processed by input mailboxes; the average amount of time taken to fully process a single email; and the number of emails that have been declared unique due to failing to read a property (such emails cannot be de-duplicated).

CA DataMinder UA Outputs

Multiple instances available; capped at 100. This contains counters relevant to outputs.

Available counters show statistics such as the number and rate at which emails are being written to the output and the average amount of time taken to write a single email to the output.

Note: The number of mailboxes supported by the Universal Adapter is limited only by the server's resources, but the number of supported performance monitor instances is capped at 100. That is, you can import from and output to as many mailboxes as necessary, but only the first hundred will be monitored using the performance counters.

Diagnostics and Error Recovery

When an email fails to be processed, the Universal Adapter periodically tries to re-process it for a configurable amount of time (see the registry values FailedRetryIntervalMinutes and TotalRetryTimeMinutes), after which it is moved to the \Failed folder. Emails can only be reprocessed from the mailbox journal Inbox.

The following sections describe how the administrator can take pre-emptive action to deal with failing emails before they create a processing bottleneck.

Note: The location of the Failed folder is specified in the FailedMailboxFolder registry value.

Move Failing Emails Immediately to the Failed Folder

The Universal Adapter can be instructed to move any **currently** failing emails to the \Failed folder immediately, without trying to re-process them. This can be useful if the emails are failing because of a data failure. That is, they will never be successfully processed due to their content. Under these circumstances, the administrator will notice that the flow of emails being processed has stopped and may want to move the emails immediately to the \Failed folder, freeing up the Universal Adapter to process other emails.

To do this, set the FailBlockedMessages registry value in the UniversalAdapter registry key to 1 and then update the registry using the UpdateConfig registry value. When the Universal Adapter has moved the e-mails to the \Failed folder, it resets the FailBlockedMessages registry value to 0.

Important! Setting the FailBlockedMessages registry value has no affect on future failing emails. It only moves currently failing e-mails.

Move Failed Emails Back to the Inbox

The Universal Adapter can be instructed to move failed emails back to the Inbox so they can be re-processed. This can be useful, for example, if a server hosting an output (such as a mailbox or folder for EVF files) becomes temporarily unavailable. If the registry value TotalRetryTimeMinutes is, say, 24 hours and the server was offline over a weekend, the \Failed folder will be populated with e-mails. The Universal Adapter will probably process these failed e-mails successfully when the output is available again, so the administrator can confidently move them back to the Inbox for re-processing.

To do this, set the RetryFAILEDFolderMessages registry value in the UniversalAdapter registry key to 1 and then update the registry using the UpdateConfig registry value (see page 24).

This moves the emails to the Inbox, and sets the RetryFAILEDFolderMessages registry value back to 0.

Important! Setting the RetryFAILEDFolderMessages registry value has no affect on future failing e-mails. It only moves emails currently in the \Failed folder.

Note: If both the RetryFAILEDFolderMessages and the FailBlockedMessages registry values are set to 1, the Universal Adapter first moves the failed e-mails back to inbox, and then moves any emails that are failing and being re-processed to the \Failed folder.

Troubleshooting

This section provides solutions to problems that may arise when you use the Universal Adapter. Note that the Universal Adapter maintains log files in its \UA\log subfolder (unless the LogFilePath registry value has been set). If problems arise when running the Universal Adapter, we recommend that you check these log files for relevant entries first.

Note: Performance monitor counters are supported for Universal Adapter inputs and outputs.

Universal Adapter Service Will Not Start

Check the Service Control Manager to ensure that the 'Log On' username and password are correct and that this account is a member of the local administrators group. Also, check that the UAMailboxName and UAMailboxServer registry values are configured correctly. We also recommend that you check the Windows Application event log for relevant entries.

Messages in My Drafts and Trash Folders Were Not Processed

The Universal Adapter is designed to process emails that have been sent. That is, emails that have been assigned a unique ID. Unsent emails such as drafts, or those moved to the Trash folder are not processed.

Error E1E15: Cannot Create a MAPI Profile

'The Universal Adapter could not create a MAPI profile to access the UA mailbox.'

This problem can occur if MS Outlook 2003 is not installed on the Universal Adapter host machine.

Error E1E1B: Universal Adapter Cannot Connect to Mailbox

'Failed to connect to mailbox <InputJournalName> with error 0xa565010f (The requested MAPI property or object cannot be found.)'

The mailbox name may be spelt incorrectly. This error can also occur if the user account under which the UA service is running does not have permission to access the mailbox. Make sure that the mailbox name is spelt correctly.

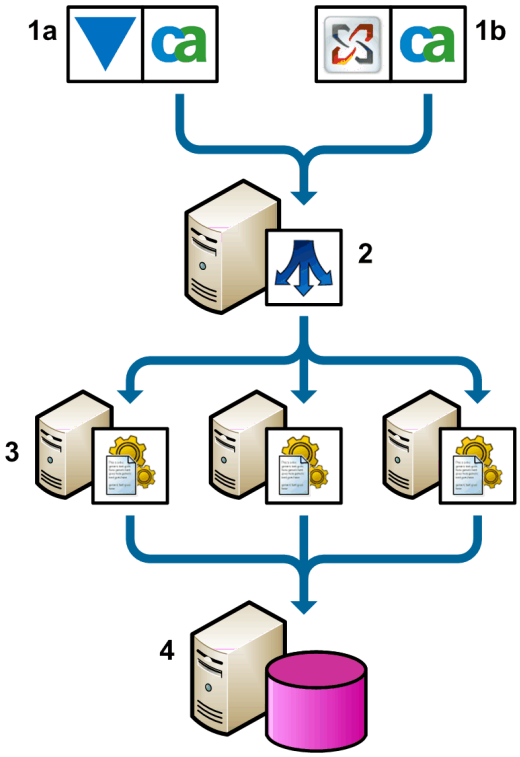
Chapter 15: Policy Engine Hubs

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.

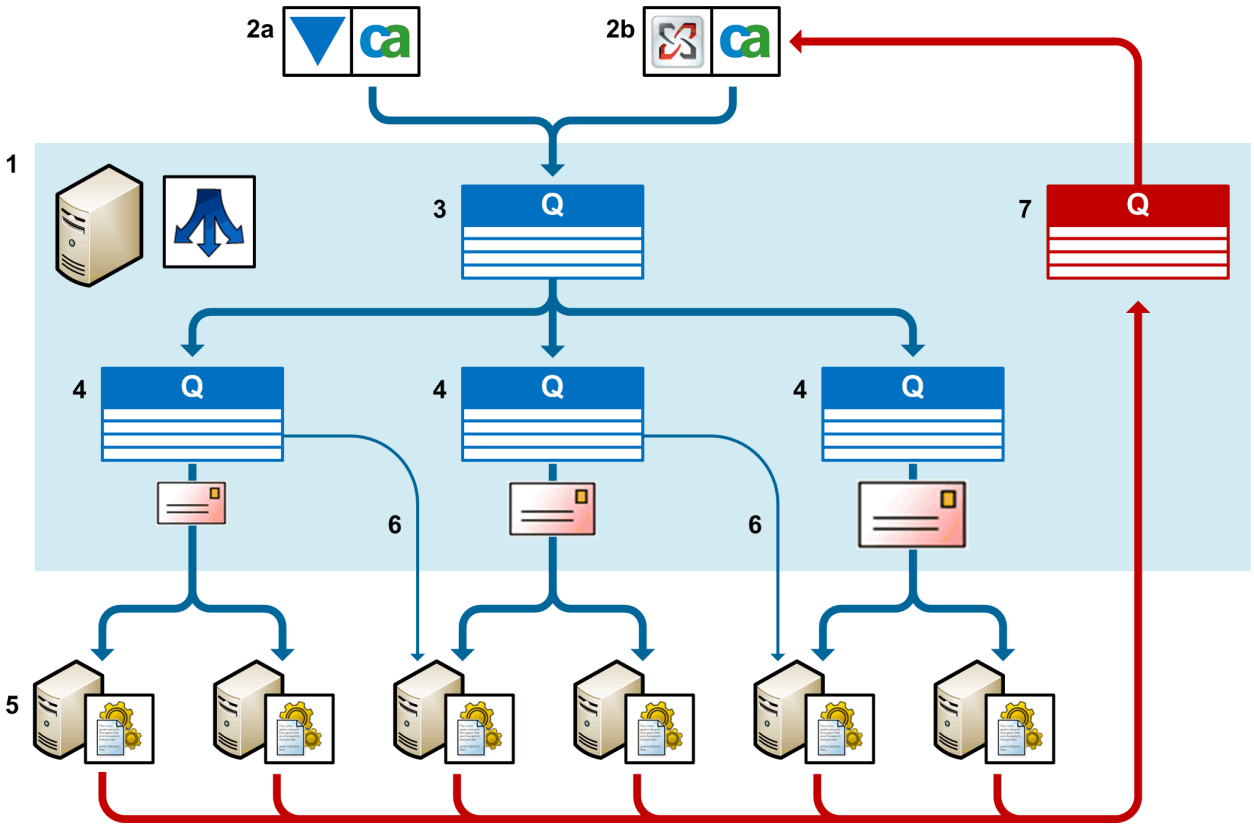


Example policy engine hub deployment

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 299) 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 299)

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

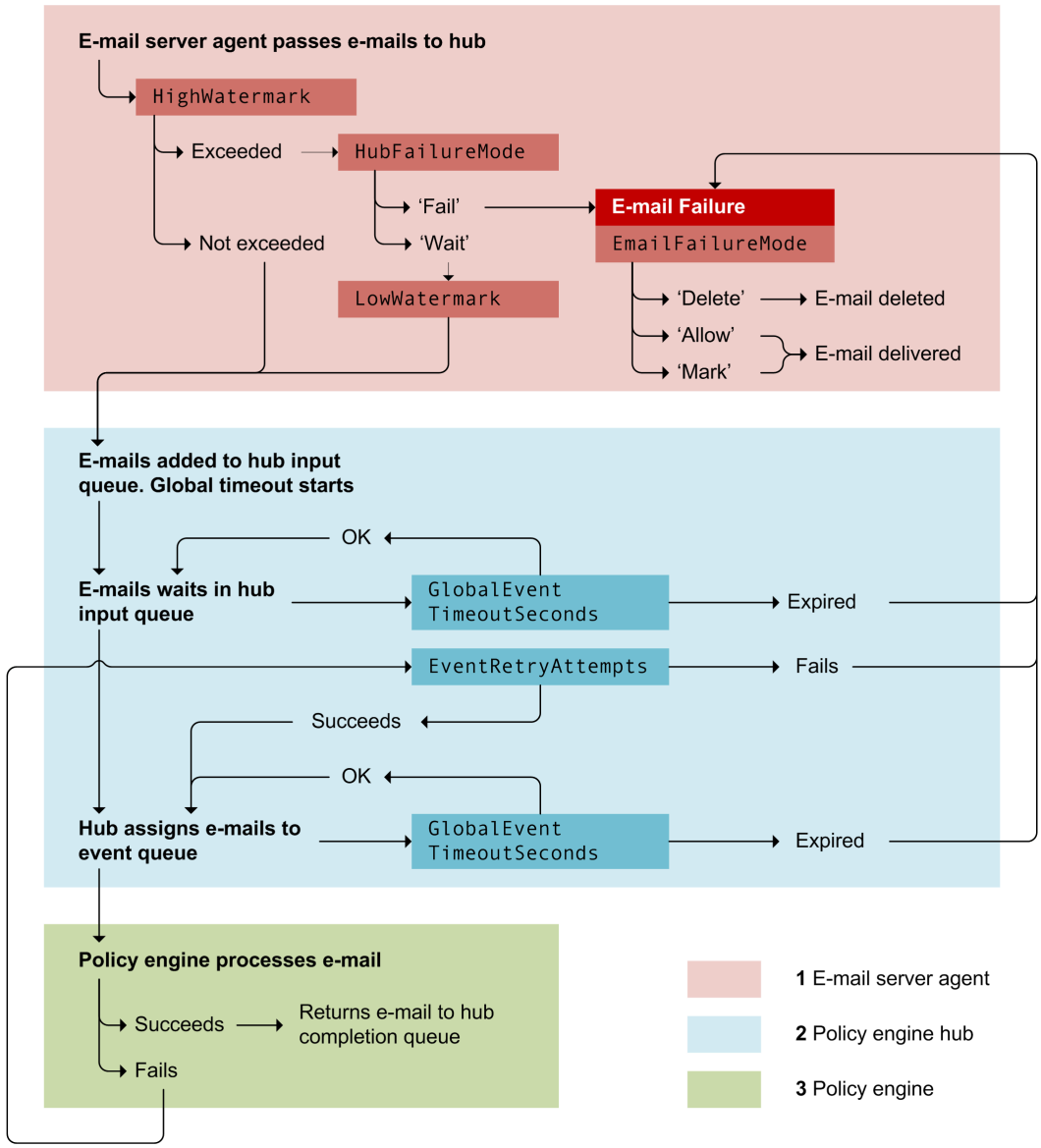
To monitor the status of individual event queues, check the relevant counters in the CA DataMinder Hub Queues performance object.

More information:

Modify the Hub Registry Values (see page 304)

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 304)

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:

Install the Policy Engine Hub (see page 302)
Configure the Policy Engine Hub (see page 302)

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:

Import Policy (see page 83)

More information:

Install the SourceOne Archive Agent (see page 182)
Install the EV Archive Agent (see page 160)
Install the ICAP Agent (see page 213)
Install the External Agent API (see page 191)

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.

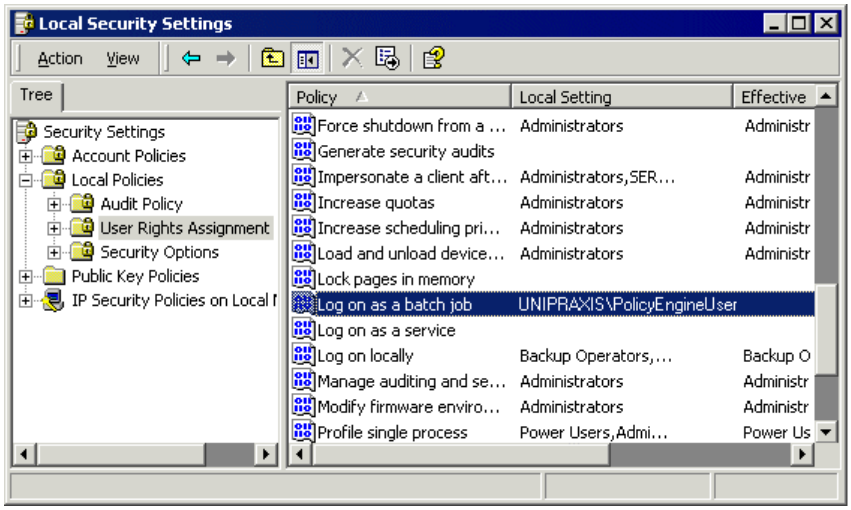
More information:

Registry Flow Chart: Email Processing on the Hub (see page 300)

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

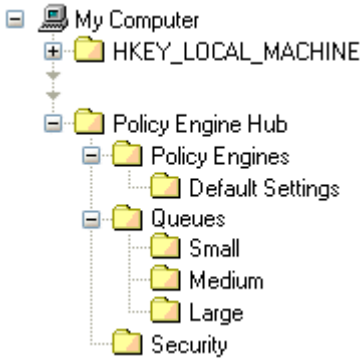
In this applet, the left pane shows the Local Policies branch and the User Rights Assignment node. The 'Log on a batch job' policy, or privilege, is shown in the right pane.

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 305)

Policy Engine Hub Registry Values

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

- Policy Engine Hub key (see page 306)
  EventLoggingLevel
  EventRetryAttempts
  GlobalEventTimeoutSeconds
  HighWaterMarkMB
  HighWaterMarkEventCount
  LogFilePath
  LogMaxNumFiles
  LogMaxSizeBytes
  LowWaterMarkMB
  LowWaterMarkEventCount
  NoPEFailTimeoutSeconds
  OperationalLoggingLevel
  PECallTimeoutMilliseconds
- DefaultSettings subkey (see page 310)
  HeartbeatPeriodMilliseconds
  MetricsPeriodMilliseconds
  ReconnectTimeoutSeconds
- Queues key (see page 312)
  ActivePolicyEngines
  AdditionalQueues
  StandbyPolicyEngines
- <Queue name> subkey (see page 313)
  ActivePolicyEngines
  MaxSizeBytes
  StandbyPolicyEngines
- Security key (see page 313)
  NTNetworkDomain
  NTNetworkUser

More information:

Modify the Hub Registry Values (see page 304)

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:

Import Failures (see page 37)
Uninstall Policy Engine Hubs (see page 318)

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 310)
<Machine name> Subkey (see page 311)

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 310)

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 297)

<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 297)

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 305)

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.

Appendix A: Accessibility Features

CA Technologies is committed to ensuring that all customers, regardless of ability, can successfully use its products and supporting documentation to accomplish vital business tasks. This section outlines the accessibility features that are supported by CA DataMinder.

Display

To increase visibility on your computer display, you can adjust the following options:

Font style, color, and size of items

Defines font color, size, and other visual combinations.

The CA DataMinder iConsole also supports a High Visibility mode. This increases the size of text and images in the iConsole screens.

Screen resolution

Defines the pixel count to enlarge objects on the screen.

Cursor width and blink rate

Defines the cursor width or blink rate, which makes the cursor easier to find or minimize its blinking.

Icon size

Defines the size of icons. You can make icons larger for visibility or smaller for increased screen space.

High contrast schemes

Defines color combinations. You can select colors that are easier to see.

Sound

Use sound as a visual alternative or to make computer sounds easier to hear or distinguish by adjusting the following options:

Volume

Sets the computer sound up or down.

Text-to-Speech

Sets the computer's hear command options and text read aloud.

Warnings

Defines visual warnings.

Notices

Defines the aural or visual cues when accessibility features are turned on or off.

Schemes

Associates computer sounds with specific system events.

Captions

Displays captions for speech and sounds.

Keyboard

You can make the following keyboard adjustments:

Repeat Rate

Defines how quickly a character repeats when a key is struck.

Tones

Defines tones when pressing certain keys.

Sticky Keys

Defines the modifier key, such as Shift, Ctrl, Alt, or the Windows Logo key, for shortcut key combinations. Sticky keys remain active until another key is pressed.

Mouse

You can use the following options to make your mouse faster and easier to use:

Click Speed

Defines how fast to click the mouse button to make a selection.

Click Lock

Sets the mouse to highlight or drag without holding down the mouse button.

Reverse Action

Sets the reverse function controlled by the left and right mouse keys.

Blink Rate

Defines how fast the cursor blinks or if it blinks at all.

Pointer Options

Let you do the following:

- Hide the pointer while typing
- Show the location of the pointer
- Set the speed that the pointer moves on the screen
- Choose the pointer's size and color for increased visibility
- Move the pointer to a default location in a dialog box

Index

F

H

I

L

M

N

P

Q

R

S

T

U

W

X

Z