Message Archiving Using the Sun Compliance and Content Management Solution

Sun Java™ Communication Suite Technical Note

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–6991–12
September 2007

070905@18135

Message Archiving Using the Sun Compliance and Content Management Solution

This technical note describes how to archive messages coming into and out of the Messaging Server using the Sun™ Compliance and Content Management Solution. Whether you require archiving for regulatory, compliance or litigation purposes, or you wish to manage the growth of your message store and reduce the storage costs, the Sun Compliance and Content Management Solution can achieve this. At this time, the Sun Compliance and Content Management Solution uses the AXS-One Records Compliance Management application. In this document, the Sun Compliance and Content Management Solution refers to the entire solution including servers and applications. AXS-One is used to specify requirements for the actual archiving and portal applications. This technical note describes how to configure the Messaging Server side. Refer to the AXS-One documentation or technical support for deployment and configuration on the AXS-One side. Note that this Technical Note assumes that a trained technical team will be helping you to size, deploy and configure this system.

This technical note contains the following sections:

- "Technical Note Revision History" on page 4
- "Archiving Overview" on page 4
- "The Sun Compliance and Content Management Solution Theory of Operations" on page 5
- "Pre-Deployment Preparation" on page 7
- "Setting Up A Compliance Archiving Deployment" on page 8
- "Setting Up An Operational Deployment" on page 12

Technical Note Revision History

Version	Date	Description of Changes
0	November 2006	Initial release of this technical note. Internal only.
1	March 2007	First external release.
2	17 April 2007	Added new information about archiving by channel.
3	5 August 2007	Added new step to operational archiving: If stubbing is required, set `MESSAGE_HASH_FIELDS` to * in `option.dat` file.

Archiving Overview

A message archiving system saves a copy of incoming and outgoing messages on a system separate from Messaging Server. Sent, received, deleted, and moved messages can all be saved in, and retrieved from, an archive system. Archived messages cannot be modified or removed by email users so the integrity of incoming and outgoing messages is maintained. Message archiving is useful for compliance record keeping, message store management, and message back up.

Archived messages can be viewed through the AXS-One portal or through Messaging Server. If the messages are deleted from Messaging Server, the AXS-One portal can be used to retrieve those deleted messages. Note, however, that archived messages are not stored in a mailbox folders structure as they are in the Messaging Server. Refer to the AXS-One literature for details.

The system can also be set up so that Messaging Server displays archived messages. For example, you can set up your system to archive messages over two years old. When the user fetches the message from a client, the message store downloads it from the archive server and displays it. From the user's point of view, the message looks the same as an normal email message. On the Messaging Server, however, instead of the message body, there is a *stub* that points to a URL of the contents of the message body stored on the archive system.

Message Archiving Systems: Compliance and Operational

There are two types of archiving, compliance and operational. Compliance archiving is used when you have a legal obligation to maintain strict retrievable email record keeping. Selected email (selected by user(s), domain, channel, incoming, outgoing and so on) coming into the MTA is copied to the archive system before being delivered to the message store or the internet. Archiving can be set to occur either before or after spam and virus filtering.

Operational archiving is used for mail management purposes. For example:

- To reduce storage usage on the message store by moving older messages to an archiving system which generally uses lower cost storage.
- As an alternative for data backup.

Note that compliance and operational archiving are not exclusive. That is, you can set up your system so that it does both compliance and operational archiving.

The Sun Compliance and Content Management Solution Theory of Operations

The interface between the AXS-One archiving system and Messaging Server consists of a shared file system called the *archive staging folder*, or simply the *staging folder*. A high level architectural view is shown below.

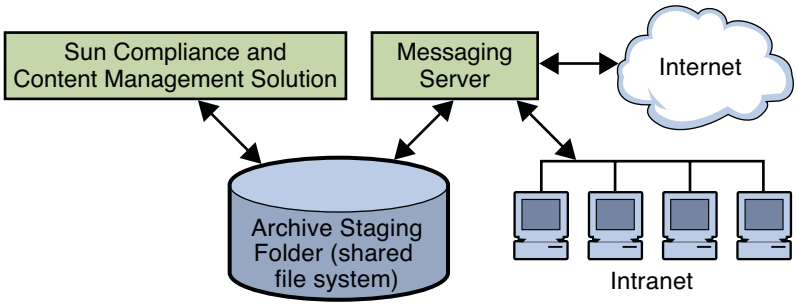

FIGURE 1 High-level Architectural View of the Sun Compliance and Content Management Solution and Messaging Server

All incoming and outgoing messages are copied to the staging folder. These messages are then moved into the Sun Compliance and Content Management Solution archive system where they can be retrieved via an AXS-One client or from the Messaging Server.

Compliance Archiving Theory of Operations

The figure below shows a low level view of a compliance architecture.

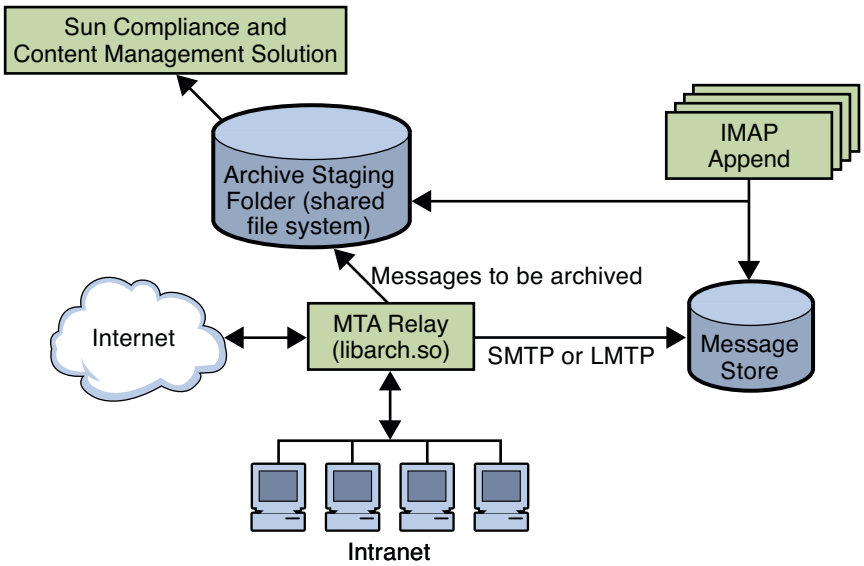

FIGURE 2 Low-Level Architectural View of the Sun Compliance and Content Management Solution /Messaging Server Compliance Archiving

As shown in the figure, messages to be archived are copied from the MTA relay to a staging folder where messages are moved into the Sun Compliance and Content Management Solution at regular intervals. Archiving can be set to occur either before or after spam and virus filtering.

An AXS-One library file called `libarch.so` is used to implement the archiving functionality on the messaging server side. The *archive stream,* that is, the messages to be archived, is controlled by the Messaging Server spam filter interface. Messages can be archived on a per user, domain, channel, or per system basis (see "Specifying the Messages to Be Filtered" in *Sun Java System Messaging Server 6.3 Administration Guide*.

The arrow pointing from the IMAP Append function to the staging directory indicates messages that are moved or copied from a non-archive folder to an archive folder are archived. This is, any new message arriving into the archived part of the system is archived.

Operational Archiving Theory of Operations

In an operational archiving deployment, messages are archived from the message store instead of the MTA. The figure below shows an architectural view of an operational archiving system.

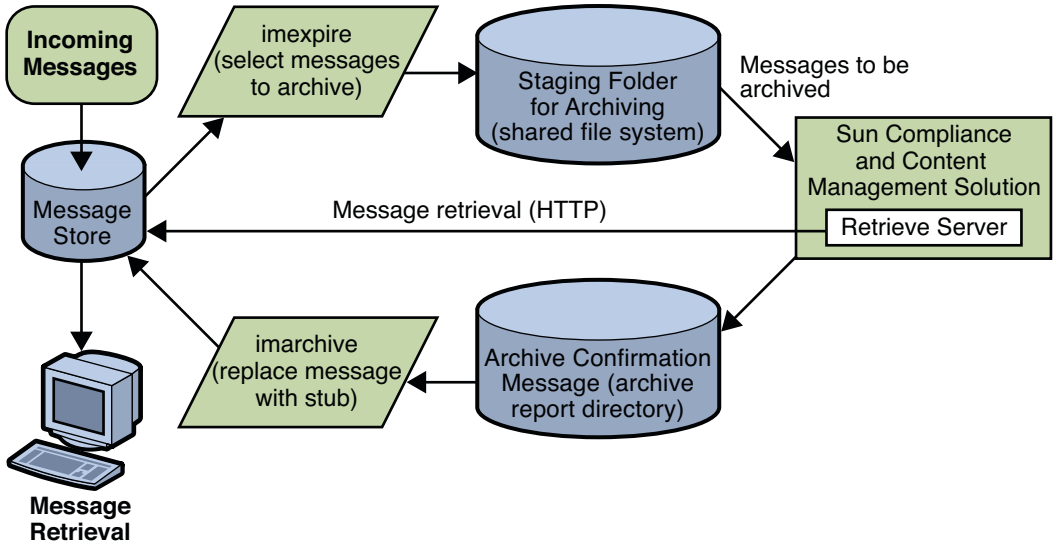

FIGURE 3 Low-Level Architectural View of the Sun Compliance and Content Management Solution/Messaging Server Operational Archiving

The diagram above shows that the `imexpire` command specifies the messages to be archived. Messages can be specified by age, size, message count, and so on (see "To Set the Automatic Message Removal (Expire and Purge) Feature" in *Sun Java System Messaging Server 6.3 Administration Guide*). These messages are copied to the staging folder where they are archived into the Sun Compliance and Content Management Solution. The AXS-One application sends an archive confirmation message to the archive report directory indicating messages that have been successfully archived. It also provides information from which URL stubs can be constructed. `imarchive` does the following:

- It marks messages in the messages store as archived and the message remains in both the message store and archive system. Marking it prevents the message from being re-archived.
- (Optional) It saves the stub of the message and deletes the RFC 822 Message.

Pre-Deployment Preparation

Before deploying an archiving system, you need to define your archiving needs and choose an appropriate architecture. See your Sun Microsystems software representative to determine the appropriate architecture.

Setting Up A Compliance Archiving Deployment

The Archive and Portal Server must be installed and configured as per the AXS-One Server documentation. It should be installed on a separate server for best performance. Messaging Server should be in full operation. Note that archiving can be done with an SMTP MMP proxy in front of the MTA host.

Compliance archiving uses the MTA's spam filter interface to specify the message stream to be archived. Be sure to understand this interface before setting up a compliance archiving system. Complete documentation is at "Specifying the Messages to Be Filtered" in *Sun Java System Messaging Server 6.3 Administration Guide*.

▼ Configuring Messaging Server for Compliance Archiving

1 **Make sure the Archive Server has been installed and configured as per the AXS-One documentation.**

2 **Make sure that the Archive and Messaging Server system users belong to the same UNIX group.**

The AXS-One UNIX system user name is `axsadm`. The Messaging Server UNIX system name is typically `mailsrv`. Both need to be in the same UNIX group. See "Creating UNIX System Users and Groups" in *Sun Java System Messaging Server 6.3 Administration Guide* for details on how to do this.

3 **Set up a shared directory for messaging archival.**

The staging directory can be a local or NFS mounted drive.

4 **Ensure that the `configutil` parameters `store.archive.compliance` and `store.archive.path` are correctly set.**

`store.archive.compliance` - Set to `ON` (default).

`store.archive.path` - Set to directory shared by the Archive Server and Messaging Server for messaging archiving in the previous step. This should be the same as the `DIRECTORY` variable in the AXS-One configuration file.

5 **Edit the `option.dat` file.**

Include following parameters:

`SPAMFILTER`*x*`_LIBRARY=/opt/SUNWmsgsr/lib/libarch.so`

`SPAMFILTER`*x*`_CONFIG_FILE=/opt/SUNWmsgsr/`*filename*

`libarch.so` is the AXS-One API for Messaging. *filename* is an arbitrarily named text file containing the AXS-One configuration information (see next step).

In both parameters, *x* is a number from 1 to 8 specifying the filtering software, in this case AXS-One. This number is used in subsequent attributes to reference the AXS-One filtering. (See step 7.)

6 Create the AXS-One configuration file. (Called *filename* in the previous step).

This file must be readable by the `mailsrv` user. It must include the following two entries:

```
STYLE=1
DIRECTORY=Message_Archival_Staging_Area
```

`STYLE` specifies the AXS-One configuration. At this time the only legal value is 1.

`DIRECTORY` specifies the directory shared by the AXS-One server and Messaging Server for messaging archiving.

7 Specify the messages to be archived.

Messages can be archived by user(s), domain, channel, incoming or outgoing mail. Use the spam filter interface to specify precisely the message stream you wish to archive. This is described in detail at "Specifying the Messages to Be Filtered" in *Sun Java System Messaging Server 6.3 Administration Guide*.

Below are some examples

Example 1. To archive all incoming and outgoing messages for users, specify an user opt-in attribute using `LDAP_OPTIN`*x* and add this attribute to your directory schema. For example:

```
LDAP_OPTIN1=AXS-One
```

Now add the opt-in attribute-value pair to the LDAP user entry of any user whose mail you wish to archive. For example:

```
AXS-One: archive
```

Example 2. To archive all incoming and outgoing messages for a particular domain, specify a domain opt-in attribute using `LDAP_DOMAIN_ATTR_OPTIN`*x*. For example:

```
LDAP_OPTIN1=AXS-One
```

Now add the opt-in attribute-value pair to the LDAP domain entry of the domains whose mail you wish to archive. For example:

```
AXS-One: archive
```

Archiving by Channel. Archiving by channel can provide greater flexibility and granularity for archiving, but it can also be a very complex process requiring deep knowledge of how messages flow between various channels and systems. Careful attention to the archiving requirements and to the email system and architecture is mandatory. Additionally, if the system is modified by adding new MTAs or channels or how channels direct message flow, then the channel archiving configuration will have to be examined.

Archiving creates a tension between keeping records of everything and avoiding the creation of unnecessary and space-wasting copies. The idea is to make exactly one copy of each message as it passes through the transport infrastructure. But in order to make just one copy and not miss anything you have to understand every possible way mail can flow.

The following two examples describe archiving by channel.

Example 3. This example shows how to archive all mail sent by local users to other local users that actually gets delivered. In this example we assume that you don't care about messages inserted directly into some other user's mailbox using IMAP, or messages sent by a local user to another local user who forwards that mail to an outside address. You don't mind if some spam gets archived occasionally. You don't care what version of a given message gets archived as long as it gets saved at some point and you'd like to avoid duplication in archiving but you don't insist on it. Finally, this example assumes that ingress and egress points follow the usual norms we recommend for messaging server setups.

Such a setup would most easily be implemented by putting a `sourcespamfilterX` (or `sourcespamfilter`*X*`optin` as required) keyword on each channel that serves as a point of ingress for local user mail and `disabledestinationspamfilterX` on each point of egress that heads out to the open Internet. (An ingress channel is the channel at which a message enters the MTA, and an egress channel is a channel at which a message leaves the MTA.)

So `sourcespamfilterX` would be put on `tcp_submit`, `tcp_auth`, and `tcp_intranet`, and `disabledestinationspamfilterX` would be put on `tcp_local`.

Typically, `tcp-local` receives inbound messages from remote SMTP hosts and outbound messages from internal users. Depending on how the system is configured, outbound messages are sent directly to remote SMTP hosts or to the smarthost/firewall system. `tcp-local` is an ingress point for incoming mail and an egress point for outgoing mail. `disabledestinationspamfilterX` will disable spam filtering if a message came from a channel that enabled spam filtering. Thus, if a message from any of the three channels with `sourcespamfilterX` goes to `tcp-local`, it will not be archived.

Example 4. This example archives all Messaging Server incoming and outgoing messages as well as internal mail. It makes the same long list of assumptions listed in Example 3above.

Put `sourcespamfilterX` on `tcp_local`, `tcp_auth`, and `tcp_submit` on inbound relays, and put `sourcespamfilterX` on `tcp_intranet`, `tcp_auth`, and `tcp_submit` on outbound relays.

8 Compile the MTA configuration with `imsimta cnbuild` followed by `imsimta restart`

MTA Options and Channel Keywords of Interest When Archiving

The `UNIQUE_ID_TEMPLATE` MTA option specifies a template used to convert an address into a unique identifier. The template's substitution vocabulary is the same as that for

`DELIVERY_OPTIONS` (see "Option File" in *Sun Java System Messaging Server 6.3 Administration Reference*). The AXS-One archiving facility will generate and use unique identifiers instead of email addresses if this option is set.

AXS-One requires the computation of a hash of each message inserted into the archive. The following MTA options control how this hash is generated:

`MESSAGE_HASH_ALGORITHM` specifies the hash algorithm. Can be any of `md2`, `md4`, `md5` (the default), `sha1`, `md128`(for RIPE-MD128), or `md160` (for RIPE-MD160).

`MESSAGE_HASH_FIELDS` — Comma separated list of fields from the header to hash (in order). Any known header field can be specified. If this option is not specified it defaults to `message-id, from, to, cc, bcc, resent-message-id, resent-from, resent-to, resent-cc, resent-bcc, subject, content-id, content-type, content-description`.

Testing Your Compliance Archiving Deployment

1. **Create two email users.**
2. **Set up your system as described in the previous section.**
3. **Send email between these users.**
4. **Verify if the staging directory contains `*.info`, `*.body.txt`, and `*.eml` files.**
   If these files exist on the staging directory, then the system is working on the messaging side.

Administering, Maintaining and Monitoring Your Compliance Archiving Deployment

- **Refer to the AXS-One documentation.**

Troubleshooting Your Compliance Archiving Deployment

1. **Make sure that the AXS-One and Messaging Server system users belong to the same UNIX group.**
2. **Verify that the spam filter interface is properly configured to capture the message stream you wish to archive.**

3. **Verify that the AXS-One Configuration file exists and is configured properly.**

4. **Check to see if there are any error messages in the** `mail.log_current`**.**

   Logging must be enabled (see "Enabling MTA Logging" in *Sun Java System Messaging Server 6.3 Administration Guide*).

5. **Verify that the staging directory has provided write permissions to the mail server user (**`mailsrv`**) and its group.**

Setting Up An Operational Deployment

Setting up operational archiving consists of two primary steps. In the first step you define what messages are to be archived using the `imexpire` command. The second step involves specifying what to do with the messages in the message store after archiving has occurred. That is, should the messages be replaced with a stub or not. This is specified using the `imarchive` command.

▼ Configuring the Messaging Server

1. **Make sure the Archive Server has been installed and configured as per the AXS-One Server Manual.**

2. **Make sure that the Archive and Messaging Server system users belong to the same UNIX group.**

   The AXS-One UNIX system user name is `axsadm`. The Messaging Server UNIX system name is typically `mailsrv`. Both need to be in the same UNIX group. See "Creating UNIX System Users and Groups" in *Sun Java System Messaging Server 6.3 Administration Guide* for details on how to do this.

3. **Set up a shared directory for messaging archival.**

   The staging directory can be a local or NFS mounted drive.

4. **Ensure that the message store** `configutil` **parameters are correctly set.**

   `store.archive.operational` - Enables operational archiving. Set to ON.

   `store.msghash.enable` - Enables message hash indexing. Must be set to ON for operational archiving.

   `store.archive.path` - This is the staging folder, the directory shared by the Archive Server and Messaging Server to pass messaging files for archiving. This is the directory defined in the earlier step.

   `store.archive.reportdir` - This is the directory used by the Archive Server to pass archiving reports back to Messaging Server. These reports are used by the `imarchive` command to

determine when it can replace the message content on an email with a URL stub. This directory is also specified on the AXS-One side and this parameter must match that value.

`store.archive.retrieveserver` - This is the fully qualified name of the Archive Retrieval Server that passes archived message text to Messaging Server or AXS-One client. This is the web server on which `SunJesRetrieveEML.asp` is enabled.

`store.archive.retrievetimeout` - Specifies how many seconds to wait for the archive server to return a message before timing out. Default is 30.

5 If stubbing is required, set `MESSAGE_HASH_FIELDS` to `*` in `option.dat` file

Example: `MESSAGE_HASH_FIELDS=*`

6 Specify the message archive policy.

Operational archiving can archive messages by folder, user, domain, number of messages in mailbox, age of messages and so on. Use the `imexpire` interface (see "To Set the Automatic Message Removal (Expire and Purge) Feature" in *Sun Java System Messaging Server 6.3 Administration Guide* to define the messages to be archived. The action attribute of the expire rules should be set to archive as follows:

```
action: archive
```

7 Use `imarchive` to specify what to do with archived messages in the message store.

After the AXS-One server archives an email message, it sends a confirmation message to the archive report directory with the document ID. At this point, the email message is safety archived and the email message on the message store can be replaced with a stub by running `imarchive`.

Be sure to run `imarchive` after AXS-One has processed all the messages. If `imarchive` is run before the messages are processed, those messages could get expired again. There are a couple of ways to ensure that this won't happen. The first is to run `imarchive` before imexpire. For example:

```
# /bin/sh
imarchive -s
imexpire
```

The other way is to simply make sure enough time has passed between running `imarchive` and `imexpire`. Use `local.schedule.`*taskname* for this (see "To Schedule Automatic Tasks" in *Sun Java System Messaging Server 6.3 Administration Guide*). In the example below, `imexpire` is run daily at 1:00am and `imarchive` is run daily at 3:00am, replacing message text with stubs:

```
configutil -o local.schedule.expire -v "0 1 * * * /opt/SUNWmsgsr/sbin/imexpire"
configutil -o local.schedule.archive -v "0 3 * * * /opt/SUNWmsgsr/sbin/imarchive -s"
```

▼ Testing Your Operational Archiving Deployment

1. **Set up the shared directories and the `configutil` variables as described in the previous section.**
2. **Create a simple test `imexpire` rule file with `archive` as the action.**

   Example:

   ```
   test.folderpattern: user/*
     test.messagesize: 1
     test.action: archive
   ```

3. **Run `imexpired –d -f` *rule_file* `-u`*testuser* and observe the output in `stdout`.**
4. **Verify if the staging directory contains `*.info`, `*.body.txt`, and `*.eml` files.**

   If these files exist on the staging directory, then the system is working on the messaging side.

5. **Invoke the AXS-One `extractor` command to process the files in the staging area.**
6. **Check the report files under the report directory.**
7. **Run `imarchive -v -s`**
8. **Validate the stubbed messages with a mail client.**

▼ Administering, Maintaining and Monitoring Your Compliance Archiving Deployment

- **See the AXS-One documentation.**

▼ Troubleshooting Your Operational Archiving Deployment

1. **Make sure that the Archive and Messaging Server system users belong to the same UNIX group.**
2. **Verify that the `imexpire` is properly configured to capture the message stream you wish to archive.**
3. **Verify that the `configutil` parameters are set correctly.**

4. **Check to see if there are any error messages in the `default` log file.**

   Logging must be enabled (see "Enabling MTA Logging" in *Sun Java System Messaging Server 6.3 Administration Guide*).

5. **Verify that the staging directory and report directory have `rwx` permissions to the mail server user (example, `mailsrv`) and the AXS-One user (example, `axsadm`).**

6. **Verify that the `imarchive` command is configured properly.**

Accessing Sun Resources Online

The web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. Books are available as online files in PDF and HTML formats. Both formats are readable by assistive technologies for users with disabilities.

To access the following Sun resources, go to `http://www.sun.com`:

- Downloads of Sun products
- Services and solutions
- Support (including patches and updates)
- Training
- Research
- Communities (for example, Sun Developer Network)

You can also view Communications Suite information at `www.sun.com/bigadmin/comms`

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 819-6991-10.