

[Back to: CUMI API Overview](#)

Contents

- [1 Summary](#)
- [2 Sample Perl Code for Forwarding a Message and Marking It for Investigative Hold](#)
- [3 Partial XML Message Schema](#)

Summary

Cisco Unity Connection provides two methods for automatically deleting messages from user mailboxes:

- Message aging determines when messages are automatically deleted for each user based on which message aging policy is assigned to the user.
- Message expiration automatically replaces .wav files with a generic "this message has expired" recording when the corresponding message has reached a specified age in days. This is a system-wide setting.

To create an archive of messages that you want to retain after they would otherwise be automatically deleted:

1. Create one or more new users whose mailboxes will be used for archiving messages. For example, you might create a second mailbox for every user whose messages you want to archive, or you might create a mailbox for an entire department.
2. Create a message-aging policy that does not age messages, and assign this policy to all of the mailboxes in which you want to archive messages past their automatic deletion date.
3. Use the CUMI API to forward the message to the applicable archive mailbox, and set the InvestigativeHold flag for the message to 1 by adding the following to the message resource:

```
<InvestigativeHold>1</InvestigativeHold>
```

Messages marked for investigative hold are not subject to message expiration.

Sample Perl Code for Forwarding a Message and Marking It for Investigative Hold

```
my $SERVER = 'cuc_server';
my $PASSWORD = 'cuc_password';
my $USER = 'cuc_user';
sub ForwardMessage {
my $recipient = shift(@_);
my $messageObjectID = shift(@_);
my $userObjectID = shift(@_);
my $subject = shift(@_);
my $credentials="$PASSWORD";
my $url=https://$SERVER/vmrest/messages?messageid=$messageObjectID&
userobjectid=$userObjectID;
my $ua = LWP::UserAgent->new;
my $header = HTTP::Headers->new;
$header->header("Content-Type", "multipart/mixed;boundary=Boundary");
my $req = HTTP::Request->new(POST => $url, $header, $credentials);
$req->authorization_basic($USER, $PASSWORD);
```

```

#Set investigative hold on/off (1=on, 0=off)
my $invHold = 1;
my $part = <<END_OF_PART;
--Boundary
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><Message>
<InvestigativeHold>$invHold</InvestigativeHold><Subject>$subject</Subject>
<ArrivalTime>0</ArrivalTime><FromSub>>false</FromSub></Message>
--Boundary
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><Recipients><Recipient><Type>TO</Type>
<Address><SmtpAddress>$recipient</SmtpAddress></Address></Recipient></Recipients>
--Boundary--
END_OF_PART
print "\nForwarding message to ", $recipient, "\n";
$req->content($part);
my $response = $ua->request($req);
};

```

Partial XML Message Schema

```

<xs:complexType name="Message">
<xs:all>
<!\- fields that can be modified by the client at any time \->
<xs:element name="Subject" type="xs:string" minOccurs="0" />
<xs:element name="Read" type="xs:boolean" minOccurs="0" />
<!\- flags that can be set by the client on send \->
<xs:element name="Dispatch" type="xs:boolean" minOccurs="0" />
<xs:element name="Secure" type="xs:boolean" minOccurs="0" />
<xs:element name="Priority" type="priorityType" minOccurs="0" />
<xs:element name="Sensitivity" type="sensitivityType" minOccurs="0" />
<xs:element name="ReadReceiptRequested" type="xs:boolean" minOccurs="0" />
<xs:element name="DeliveryReceiptRequested" type="xs:boolean" minOccurs="0" />
<xs:element name="InvestigativeHold" type="xs:boolean" minOccurs="0" />

```

Back to: [CUMI API Overview](#)