HP MAILbus 400 Message Transfer Agent and Application Program Interface

Release Notes for OpenVMS

October 2006

Revision/Update Information:	This is a revised manual.
Operating System and Version:	OpenVMS Alpha Version 7.3-2 and above OpenVMS VAX Version 7.3
Software Version:	HP Mailbus 400 Message Transfer Agent V3.2-12 for OpenVMS

Hewlett-Packard Company
Palo Alto, California

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Printed in the US

The HP OpenVMS documentation set is available on CD-ROM.

This document was prepared using DECdocument, Version 3.3-1B.

Contents

Part I Introduction

Part II Installing Version 3.2-12 and Fixes included in this release

Part III Restrictions and Documentation Errors

Part IV Background to Previous Kits

Tables

Part I

Introduction

This part provides a brief introduction to the products and describes the structure of these Release Notes.

1 Introduction

These Release Notes provide information relating to both the MAILbus 400 Message Transfer Agent (MTA) and the MAILbus 400 Application Program Interface (API) on OpenVMS.

Unless otherwise stated, the information in these Release Notes applies to the MTA and API running on both OpenVMS Alpha and OpenVMS VAX.

1.1 MAILbus 400 MTA and API

The MAILbus 400 MTA provides X.400 message transfer services User Agents and Gateways.The MAILbus 400 MTA is an MTA conforming to the CCITT 1988 X.400 Recommendations and International Standard ISO/IEC10021, plus the revisions and recommendations that form the 1992 editions of the X.400 standards.

The MAILbus 400 API is a callable interface that can be used to build applications that access the MAILbus 400 MTA and use its message transfer service.

2 Structure of the Release Notes

These Release Notes are divided into four parts:

- Part I provides a brief introduction to the products and describes the structure of these Release Notes.
- Part II describes the problems fixed in this release and changes introduced in Version 3.2-12 of the MAILbus 400 MTA. There are no changes to the MAILbus 400 API in this kit.
- Part III lists the restrictions and documentation errors that apply to this release and previous kits.
- Part IV provides information about documentation and interworking kits. It includes the following:
  - The documentation provided with the products
  - Interworking information

Part II

Installing Version 3.2-12 and Fixes included in this release

This part is divided into three main sections, as follows:

- Section 3 describes how to install Version 3.2-12 of the MAILbus 400 MTA and API.
- Section 4 describes the problems present in V3.2 fixed in this Version 3.2-12.
- Section 5 describes the differences between Version 3.2 and Version 3.0 of the MAILbus 400 MTA.

3 Installing Version 3.2-12

To install this kit follow the instructions, with the exceptions listed below, in the relevant installation guide:

- *HP MAILbus 400 MTA Installing on an OpenVMS Alpha System*
- *MAILbus 400 API Installing on an OpenVMS Alpha System*

The exceptions to the installation instructions are:

- Make sure you install one of the following configurations of prerequisite software:
  - OpenVMS Alpha V7.3-2
    HP DECnet-Plus for OpenVMS V7.3-2 including the DECnet Application Interface component.
    HP Enterprise Directory Service for OpenVMS V5.3 or later.
  - OpenVMS Alpha V8.2
    HP DECnet-Plus for OpenVMS V8.2 including the DECnet Application Interface component.
    HP Enterprise Directory Service for OpenVMS V5.4 or later.
  - OpenVMS Alpha V8.3
    HP DECnet-Plus for OpenVMS V8.3 including the DECnet Application Interface component.
    HP Enterprise Directory Service for OpenVMS V5.4 or later.
- The approximate disk space required on the system disk (in blocks) during and after installation for the different components are shown in the following tables:

Table 1 Disk Requirements on OpenVMS Alpha

Component Title	Space Required (in blocks)	
	During	After
MAILbus 400 MTA Mgt	4300	2400
MAILbus 400 MTA Base	5300	4400
MAILbus 400 MTA Server, Mgt, and Base	30000	28000
MAILbus 400 API and Base	5300	4600

The command to install either the MTA or the API is:

```
$ @SYS$UPDATE:VMSINSTAL MTA032 device-name:
[MTA032]OPTIONS N
```

The image identification of the new images in this kit is MTA 3.2-12.

The version number of the kit when displayed using NCL management is 3.2.12.

4 MTA Problems Fixed in this release

4.1 Missing Domain Information

When a peer MTA association gets interrupted, domain information does not get recovered. This leads to the domain information not getting logged in the accounting file for the subsequent messages transferred over the recovered association. This problem is fixed in this release.

4.2 MTA crash while extracting filename extension

MTA crashes while extracting the filename extension from an FTBP message because of improperly terminated string in extension. This problem is fixed in this release.

4.3 MTA crash in the Relayer region

MTA crashes while logging information about a structure in the MTA log file. The crash occured in the DECCRTL fprinf routine because incorrect parameters were getting passed. This problem is fixed in this release.

4.4 MTA events were logged with incorrect information

In cases where a manually configured peer MTA entity is not present, or is incorrectly configured, the Inbound Hard Rejection events logged by the MTA indicated a peer MTA entity of type "Automatically Configured" even when the peer MTA was "Manually Configured". This problem is fixed in this release. Now, the events log the peer MTA type as "Not Configured" in such cases.

4.5 MTA crash

MTA crashes while posting an Inbound connection failure event. This problem is fixed in this release.

5 Differences Between Version 3.2 and Version 3.0

5.1 MTA Revised Filter Mechanism

This release of MTA provides a revised filter mechanism for restricting the delivery and relay of mails to user agents/gateways or peer MTA's. The filter can be applied based on the filter settings at the recipient O/R address level.

The Filtering mechanism provides the ability for a User to define an "Authorized Originators" and "Denied Originators" list of Originators Partial or Complete Oraddress at the recipient Oraddress level. The filter behavior is governed by "Originator Restriction" attribute. The filter mechanism and the associated NCL directives are explained in the NCL online help.

The Revised Filter attributes override the filter mechanism introduced in MTA V3.0, described in Section 5.1 and 5.2.

5.2 CDIF Converter Crash Problem is Resolved

If an FTBP bodypart was preceded by a forwarded message, the successfully converted BP14 bodypart for the FTBP, was lost while re-packaging the same into the final message for delivery. This problem and the resulting CDIF converter crash problem has been resolved.

5.3 Retry Interval for Agent Made Configurable

The Retry Interval for Agent has been implemented by means of a System-wide logical, namely 'MTA$AGT_RETRY'. If this logical is not set, then the timer value would be taken from the default compile time parameter of TEN minutes.

The logical 'MTA$AGT_RETRY' has to be defined system-wide before the MTA Startup.

```
$ DEFINE/SYS MTA$AGT_RETRY "<time-factor-in-minutes>"
```

For example,

```
$ DEFINE/SYS MTA$AGT_RETRY "20"
would assign 20 minutes to the Retry Interval timer.
```

5.4 Comparison of Routing Instruction Attributes, Agents and Domains has been made Case-insensitive

During the routing of messages, a case-sensitive comparison was made for agent names and domain names. This has been modified to a case-insensitive comparison.

5.5 Logical for Enabling/Disabling STA Functionality

A system-wide logical, namely 'MTA$STA' has been introduced in the MTA which will allow the customer to turn ON/OFF the STA from the MTA at the time of MTA startup.

```
$ DEFINE/SYS MTA$STA "<value>"
```

If STA needs to be enabled, then the logical has to be set to a value of "TRUE", for example,

```
$ DEFINE/SYS MTA$STA "TRUE"
```

The creation of the STA threads in the MTA is dependent on the value "TRUE" assigned to the Logical. If ANY OTHER value is assigned or the logical has not been defined at all, then the STA threads would not get created in the MTA.

5.6 NCL Output Buffer Size Increased to 64K

The NCL output buffer size has been increased to 64K. This allows user to view large number of entries in the NCL output.

5.7 SSRT3624 X.400 Potential Security Vulnerability via ASN.1 Cross Reference: NISCC (006489)

A potential denial of service had been identified that may allow a remote initiated Buffer Overflow when malformed ASN.1 messages are submitted. This potential buffer overflow has been fixed in this version.

5.8 Conversion Failures When MTA Tries to Convert a Message Containing Euro Character

When MTA tries to carry out a conversion of the message sent by the Microsoft based P7 User agent containing the Euro character, the conversion fails. Also, the associated error logs do not contain sufficient information to deduce the cause of the failure. This problem has been fixed in this release.

5.9 Fix for MTA Crash While Processing a Particular Message

MTA was crashing while processing a message containing an FTBP bodypart and one of the file attributes parameter (extension field) of this bodypart is of unusual length. This problem has been fixed in this release.

5.10 "No such Entity" Problem. Fix for MTA Crash

MTA was crashing randomly when CDIF conversion was enabled when processing forwarded messages containing FTBP bodypart.

5.11 NCL Commands Reporting "Noresource Available" Error

NCL commands were reporting the above error when the buffered I/O quota for the MTA$MTS process gets exhausted. This has been fixed in this release.

5.12 MPDUs in "Awaiting Processing Retry" State

When a message without any subject and bodypart arrives from an external domain to an MTA and inserting warning text feature is enabled, the MPDU thus transferred would be in "Awaiting Processing Retry" state. This problem has been fixed in this release.

5.13 The suggestions that were provided on the test kit with Revised Filter functionality have been incorporated in this release

The Error text of Oraddress Filter Rejection Event has been modified as below:

```
Originator Restriction = Authorize
Failure would imply Denied Originators Address match hence
Error=Originator is denied to send to recipient

Originator Restriction = Deny
Failure would imply missing Authorized Originators Address hence
Error=Originator not authorized to send to recipient
```

Filter Events will be blocked by default during startup.

Part III

Restrictions and Documentation Errors

This part lists the restrictions and documentation errors that apply to Version 3.2-12 and previous kits, as follows:

- Restrictions within HP DECnet-Plus that affect the MAILbus 400 MTA (see Section 6).
- Product Restrictions (see Section 7).
- Errors in the Documentation (see Section 8).

6 Restrictions in DECnet-Plus/NCL/Enterprise Directory Service for OpenVMS

The MAILbus 400 MTA is an application layered on HP DECnet-Plus. Read the HP DECnet-Plus Release Notes to be aware of any restrictions in HP DECnet-Plus that might have an impact on the MAILbus 400 MTA.

The following section lists an additional restriction in HP DECnet-Plus which affects the management of the MAILbus 400 MTA.

6.1 New attributes of the MTA V3.2 are unavailable due to DECnet-Plus/Enterprise Directory Service Restrictions

The following two subsections describe the restrictions with DECnet-Plus/Enterprise Directory Service due to which the new MTA V3.2 attributes become unavailable via NCL.

6.1.1 DECnet-Plus Restrictions

The new attributes of MTA V3.2 will be included into future releases of DECnet. Till then, DECnet engineering has provided a test NCL$GLOBAL_SECTION.DAT to be used with this kit.

Please backup the NCL$GLOBAL_SECTION.DAT that is presently being used on your system, and copy the one that is being provided to SYS$LIBRARY. You will need to reboot the system. This should enable the use of new filter attributes.

6.1.2 Enterprise Directory Service Restrictions

Due to limitations in the schema supplied with HP Enterprise Directory for e-Business V4.0-25 to V5.2, not all the new features of the MTA V3.2 are available with these versions of the Directory.

If this is the case, you can update the schema by manually re-compiling it. The schema files which contains the Revised Filter attributes is provided along with the kit. Section 14.1 describes manual compilation of the schema. Al can upgrade to a higher version of the HP Enterprise Directory for e-Business. You will then be able to use all the new features of the MTA V3.2.

6.2 NCL Restrictions

The following subsections describe restrictions with NCL that have an impact on the MAILbus 400 MTA.

6.2.1 Limited NCL Command Buffer Size

The NCL command input buffer size is 2048 bytes. This limits the number of characters that can be entered in one NCL command. When you interactively enter an NCL command of more than 2048 bytes, the command fails.

This restriction is especially relevant when you use the MTS module to create directory entries that have long attribute values, for example long distribution lists. You are advised to create a distribution list with some members and subsequently add additional members to it using the ADD command, for example:

```
CREATE MTS "/MTS=ACME" ORADDRESS -
   "C=NZ;A=NZ-PTT;P=ACME;O=ACME;CN=DIST-LIST1" -
   TYPE=DISTRIBUTION LIST, MEMBERS { -
   "C=NZ;A=NZ-PTT;P=ACME;O=ACME;OU1=WELL;CN=Clare Roberts", -
   "C=NZ;A=NZ-PTT;P=ACME;O=ACME;OU1=AUCK;CN=William Laurence", -
   "C=NZ;A=NZ-PTT;P=ACME;O=ACME;OU1=WELL;CN=Jane Underwood", -
   "C=NZ;A=NZ-PTT;P=ACME;O=ACME;OU1=AUCK;CN=Fred Smith" }
```

```
ADD MTS "/MTS=ACME" ORADDRESS -
   "C=NZ; A=NZ-PTT; P=ACME; O=ACME; CN=DIST-LIST1"  -
   MEMBERS { -
   "C=NZ;A=NZ-PTT;P=ACME;O=ACME;OU1=WELL;CN=Henry Petra", -
   "C=NZ;A=NZ-PTT;P=ACME;O=ACME;OU1=WELL;CN=Simon Prior", -
   "C=NZ;A=NZ-PTT;P=ACME;O=ACME;OU1=WELL;CN=Freda Turner", -
   "C=NZ; A=NZ-PTT;P=ACME;O=ACME;OU1=WELL;CN=Cathy Barnes" }
```

6.2.2 Showing Write-Only Characteristic Attributes Using NCL

If you use NCL to show write-only characteristic attributes of the MTA or MTS modules, for example, the Password attribute, you will receive the following response:

```
Command failed due to:
 get list error
```

Ignore this response.

6.2.3 NCL and Quoted Strings

Do not split quoted strings across more than one line when using NCL. The following is a correct example of an NCL command containing quoted strings:

```
CREATE MTS "/MTS=ACME" ORADDRESS -
 "C=NZ;A=NZ-PTT;P=ACME;O=ACME;OU1=WELL;CN=Clare Roberts" -
ROUTING INSTRUCTION [ACTION=DELIVER,SERVER MTA="WELL.MTA-NODE6", -
 AGENT="UA1", DEFINITIVE ORADDRESS = -
 "C=NZ;A=NZ-PTT;P=ACME;O=ACME;OU1=WELL;CN=Clare Roberts"]
```

6.2.4 Problems Showing MPDUs With a Large Number of Recipients

As a result of a restriction within HP DECnet-Plus, the following command fails when an MPDU has more than about 100 recipients:

```
SHOW MTA MPDU * RECIPIENTS
```

6.2.5 Cluster-Wide NSAPs

You can set up more than one MTA in a cluster. However, each MTA is independent, that is, the MTAs do not share workspaces and operate as they would on individual systems.

As part of setting up the MTA, the Presentation address, and where applicable the Session address, for the MTA are automatically added to the MTA's entry in the directory. As a result, all the NSAPs available on the system where the MTA is set up are added to the MTA's addresses. If the system is part of a cluster, and there is a cluster alias, there might also be a cluster-wide NSAP. This NSAP will also be included in the MTA's addresses, but will not be used by any MAILbus 400 MTA within your routing domain.

When you provide the MTA's Presentation or Session address to managers of peer MTAs in other routing domains, do not include the cluster-wide NSAP, or these MTAs will not be able to make connections to the boundary MTA.

To find out if there is a cluster-wide NSAP on the system, type the following command:

```
NCL> SHOW OSI TRANSPORT LOCAL NSAP * NSAP ADDRESS
```

The response to the command lists all the NSAP addresses on the system. Look for an NSAP address that is identified by the cluster alias name rather than the node name.

6.2.6 OSI Applications Need to be Restarted When the OSI Transport Entity is Deleted

When the OSI Transport entity is deleted (possibly as part of a HP DECnet-Plus shutdown) any Transport Template entities created for the MTA are also deleted. Without these Transport Template entities, the MTA cannot make outgoing Transport connection requests. Furthermore, the MTA is no longer listening for incoming connections from OSI Transport.

If you suspect that the OSI Transport entity has been deleted, disable and delete the MTA by executing the MTA shutdown procedure SYS$STARTUP:MTA$COMMON_SHUTDOWN.COM. You also need to stop OSAK and the HP Enterprise Directory Service.

After OSI Transport has been restarted, start OSAK and the HP Enterprise Directory Service. You then need to execute the MTA startup procedure MTA$COMMON_STARTUP.COM, located at SYS$STARTUP.

Executing these procedures ensures that all required entities, including the Transport Template entities for the MTA, are created.

6.2.7 OSI Transport Entity Attribute: CONS NSAP Addresses

Make sure that you do not provide a CLNS NSAP address as a value for the CONS NSAP Addresses attribute of the OSI Transport entity. You must add a CONS NSAP in the CONS NSAP Addresses attribute.

7 Product Restrictions

The following sections describe restrictions in the MAILbus 400 MTA and MAILbus 400 API that you need to be aware of.

7.1 Restrictions in the MAILbus 400 MTA

The following sections list known restrictions in the MAILbus 400 MTA.

7.1.1 Teletex Characters in Routing Domain Names

You cannot use teletex characters in any part of the distinguished name that identifies the routing domain entry in the directory. As an example, you cannot enter `/C=AU/O=MY_ORG/MTS=ACME`, because the underscore (_) is a teletex character.

If you do enter teletex characters in the distinguished name for the routing domain entry, the MTS entity returns the following message:

```
command failed due to:
 process failure
```

7.1.2 Re-Instating MTS and MTA Online Help Following a DECnet Upgrade

If you upgrade DECnet and find that the MTA and MTS Online Help are no longer accessible, you will need to re-install the MTA Management component of the MAILbus 400 MTA. Follow the instructions in the *HP MAILbus 400 MTA Installing on an OpenVMS Alpha System* or *HP MAILbus 400 MTA Installing on an OpenVMS VAX System* and select MAILbus 400 MTA Mgt.

The MTA and MTS Help files will then be added to the NCL Help library.

7.1.3 Label for Extended Network Address

There is a restriction in the MTS entity that means you cannot enter PSAP as the label for an Extended Network Address.

7.1.4 MTA Requires Unique PSEL, SSEL or TSEL Values for Inbound Communication Over CONS

This restriction only applies to a boundary MTA that makes connections to peer MTAs in other routing domains, that is, peer MTAs for which you have created Peer MTA entities.

When validating an incoming connection from a peer MTA, the MAILbus 400 MTA uses the peer MTA's Presentation or Session address. If the peer MTA is in another routing domain, the MAILbus 400 MTA selects a Peer MTA entity representing the peer MTA on the basis of the Presentation or Session address attributes held in the Peer MTA entity representing the peer MTA.

If you set up multiple Peer MTA entities where the CONS NSAP is the only difference in the Presentation or Session address attributes, the MAILbus 400 MTA will consider them to be identical, and will then choose an entity based on other attributes, for example Direction, Peer Name or Peer Password.

7.1.5 Activity Name Not Specified in Events

The name of the Activity is incorrectly omitted from the entity name in following events:

- Inbound Failure
- Outbound Soft Rejection
- Outbound Hard Rejection
- Outbound Establishment Failure
- Outbound Failure
- Lower Layer Protocol Violation
- RTSE Protocol Violation

The following is an example of the Outbound Failure event with the missing Activity name:

```
Event: Outbound Failure from: Node ACME:.NODE2  MTA Peer MTA
       [
       Type = Manually Configured ,
       Name = "mta-node3"
       ] Activity ,
        at: 1994-08-11-04:51:36.283+01:00I2.230
        RTSE Entity Initiating Failure=Peer MTA,
        Abort Reason Code=Permanent Problem,
        Local Diagnostic=No Diagnostic Available,
        Octets Out=46366,
        MPDUs Out=96
        eventUid   70246A56-95F6-11CC-96D1-AA00040066AA
        entityUid  43F966FE-95F4-11CC-9679-AA00040066AA
        streamUid  D7C72E60-8B19-11CC-8005-AA00040066AA
```

7.1.6 Disabling Peer MTA Entities During Message Transfer

You are advised not to disable a Peer MTA entity, or delete an associated Activity entity, if there is data being transferred to or from the peer MTA represented by the Peer MTA entity. The exception to this is if you need to delete an Activity entity because there is a problem with data transfer across the association represented by the Activity entity.

You can determine whether data is being transferred by checking the Activity entities for a peer MTA. Enter the following command:

```
NCL> SHOW NODE MTA PEER MTA identifier ACTIVITY * STATE
```

where *identifier* is the identifier for the peer MTA.

If the response indicates that there is one or more Activity entity in the state Active, the MTA is transferring data to or from the specified peer MTA.

If you do disable a Peer MTA entity while data is being transferred, or delete Activity entities that are in the state Active, the MTA might stop operating.

7.1.7 Cannot Use the WITH Clause in MTA and MTS Module Commands

You cannot use the WITH clause in an MTA or MTS Module NCL command. If you do attempt to use the WITH clause, a wildcard search is completed instead.

7.1.8 Using MTA and MTS Module SHOW Commands With Repeated Attributes

You cannot enter a SHOW command that includes ALL and other named attributes in the command, as shown in the following examples:

```
NCL> SHOW MTA MPDU * ALL ATTRIBUTES,NAME
NCL> SHOW MTA CREATION TIME, ALL COUNTERS
```

If you use the SHOW command as described in these examples, you receive the following error:

```
command failed due to:
 process failure
```

You also receive the following System Interface Error event:

```
Event: System Interface Error from: Node ACME:.NODE1 ,
        at: 1994-06-14-10:09:05.526+01:00I0.420
        System Interface Error=OpenVMS Management Operation,
        Parameter="EMAA read",
        Error Text="no such file or directory"
        eventUid   A7318173-8760-11CD-9E07-AA000400FEFF
        entityUid  296404B3-875C-11CD-9D9D-AA000400FEFF
        streamUid  AEF0C4FC-7DFA-11CD-8016-AA000400FEFF
```

7.1.9 Maximum Attribute Length Not Checked for Directory Entries

No checks are made on the maximum length of attribute values supplied when directory entries are created using the entities of the MTS module.It is therefore possible to create entries in the directory whose attributes might not be acceptable according to CCITT requirements. The use of unacceptably long attributes might result in problems when the MAILbus 400 MTA is interworking with messaging systems that are implemented strictly according to the CCITT X.400 Series of Recommendations.

When creating entries in the directory, follow the attribute length restrictions indicated in the MAILbus 400 MTA documentation.

7.1.10 MTS Module and Long Distribution Lists

If you enter a SHOW command for a long distribution list, for example a distribution list with more than 130 members, the NCL prompt is not returned.

If you suspect that you have a problem with the MTS entity as a result of showing a long distribution list, stop the MTA$MTS process on the system where you entered the command. To stop and restart the MTA$MTS process, complete the following steps:

1. Find out the process identifier (PID) for the MTA$MTS process as follows:

   ```
   $ SHOW SYSTEM
   ```

2. Identify the PID associated with the MTA$MTS process and use the following command to stop the process:

   ```
   $ STOP/ID=pid
   ```

   where ***pid*** is the PID for the MTA$MTS process.

3. Restart the MTS process as follows:

```
$ @SYS$STARTUP:MTA$MTS_INIT
```

7.1.11 Access to Accounting Log Files

Applications cannot access the current accounting log file until after it is closed by the MTA and a new file opened. A new file is opened every four hours.

7.2 Restrictions in the MAILbus 400 API

This section lists known restrictions in the MAILbus 400 API.

7.2.1 Use of `ma_wait` and `mt_wait`

When awaiting a response from the ma_wait or mt_wait routine, the client is *not* notified if either of the following occurs:

- A message changes state from that of reserved to unreserved.

  This restriction applies to the ma_wait and the mt_wait routines.

- A message becomes available after being temporarily unavailable.

  This restriction applies only to the ma_wait routine.

7.2.2 Remote Client Connections Using CONS

You cannot use a CONS NSAP for remote client connections over OSI Transport for this version of the MAILbus 400 API; you are restricted to using a CLNS NSAP or TCP/IP NSAP.

7.2.3 No Support for Multi-Threaded Environments

The MAILbus 400 API is not supported in a multi-threaded environment.

7.2.4 Using `om_encode` and `om_decode` to Encode and Decode Objects

You can only use `om_encode` or `om_decode` to encode and decode objects of the following classes:

- The OR Address and OR Name classes within the MH Package.
- The External class within the OM Package.

You cannot use `om_encode` and `om_decode` with any class within the IM Package.

7.2.5 Mandatory Fields on Delivery Envelope for Forwarded Message

In line with the X/Open™ specification for X.400 mail, the MAILbus 400 API allows an application to omit the attributes Actual Recipient Name, Content Type, Originator Name, and Submission Time from the Delivery Envelope of a forwarded message. According to X.420 and X.411 this is incorrect.

An application should ensure that these fields are present on a forwarded message.

7.2.6 OM_NETWORK_ERROR When Application Passes Invalid Data

Chapter 5 of *MAILbus 400 API Programming* explains what the API Server is. If your application uses the API Server to connect to the MAILbus 400 MTA, passing incorrect data to the MAILbus 400 API can lead to an OM_NETWORK_ERROR error.

7.2.7 `om_instance` Only Works for Service-Generated Objects

The OM routine `om_instance` only determines the instance of a service-generated object, either public or private. It does not work for client-generated objects.

7.2.8 OSI Transport Not Available if You Link Against Archive Libraries

If you link your application against archive libraries, the application cannot use OSI Transport to connect to the MAILbus 400 MTA. See Chapter 6 of *MAILbus 400 API Programming* for further information about linking.

7.2.9 Single-Process MAILbus 400 Application Cannot Use XDS if Linked Against Archive Libraries

A single-process application cannot use both the MAILbus 400 API and the API to the HP X.500 Directory Service (XDS), if the application is linked against archive libraries. An application that wishes to use both the MAILbus 400 API and XDS must either use shared libraries or have multiple processes and keep calls to the two interfaces in separate processes.

See Chapter 6 of *MAILbus 400 API Programming* for further information about linking.

7.2.10 1992 Changes Not Implemented in MAILbus 400 API

The MAILbus 400 documentation set refers to the new ITU–TS (formerly "CCITT") recommendations as "the 1992 Standards". The MAILbus 400 API does not implement all the changes introduced since the 1988 publication. The following sections explain how MAILbus 400 API does not implement the 1992 Standards completely.

- Other Notifications

One of the additions to the standards since 1988 is the ability for IPM contents to carry a notification other than the two defined in the 1984 standards (Receipt Notification and Non-receipt Notification). The new notifications are user-definable and called “Other Notifications”. Any user-defined notification that a MAILbus 400 MTA passes to an API application is presented in its encoded form in a General Content object.

- File Transfer bodypart

  One of the additions to the standards since 1988 is the IPM File Transfer bodypart. The MAILbus 400 API does not support a File Transfer bodypart object class.

  To avoid your application receiving File Transfer bodyparts, ensure that the Content Information for your users’ O/R addresses is Content Type 1992 Externally Defined IPMS, that is, the object identifier, "{1 3 12 2 1011 5 5 0 1 22}". Using this content type will make sure that the MTA translates File Transfer Bodyparts to Externally Defined Bodyparts.

- IPM Extensions

  There is no support in the MAILbus 400 API for the IPM extensions introduced in the 1992 MHS Standards. These are:

  the `auto-submitted-indication` field
  the mechanism for handling unknown extensions

- P1 Per-Message Flags

  The 1992 MHS Standards defined two additions to the P1 Per-Message Flags. These are not accessible using the MAILbus 400 API.

- New Diagnostic Codes

  The 1992 MHS Standards introduce some new reason and diagnostic codes for reports. It is possible that an application may receive these codes, but there is no symbol defined for them in the MAILbus 400 API’s header files.

8 Errors in the Documentation

The following section lists errors in the documentation provided with the MAILbus 400 API.

8.1 MAILbus 400 API Documentation

The following sections describe errors in or additions to the MAILbus 400 API documentation.

8.1.1 Error in Section 3.3.1 of Programming Guide

There is an error in the second paragraph of Section 3.3.1 of *MAILbus 400 API Programming*. Replace the last two sentences of the second paragraph in Section 3.3.1 with the following:

If there is a Definitive O/R address in the routing instruction, the API uses this Definitive O/R address for the Distinguished Recipient Address attribute on objects of the Delivery Envelope. If there is no Definitive O/R address in the routing instruction, the Distinguished Recipient Address attribute is not present. In Delivery Reports, the Distinguished Recipient Address attribute is always present.

8.1.2 Addition to Internal Trace Entry Description in Chapter 8 of Programming Guide

There is some information missing in the Internal Trace Entry description on page 8-67 of *MAILbus 400 API Programming*. Add the following text below the Attributes table on page 8-68:

The Attempted Country Name, Attempted ADMD Name, and optionally the Attempted PRMD Identifier OM attributes make up the Attempted Global Domain Identifier. The Internal Trace Entry should contain either an Attempted MTA Name or an Attempted Global Domain Identifier, and not both.

8.1.3 Correction to Description of `mtX_open`

The description of the Password argument of the `mtX_open` MT routine on page 14-12 of *MAILbus 400 API Programming* should read as follows:

The password associated with the Client, as specified when the Client application was registered with the MTA. You should indicate the absence of a password by using OM_STRING with a null pointer of zero length.

8.1.4 Multi-Valued Attributes

There are certain attributes, in an object of the OR Address class, that can have more than one value; values of these attributes can have both a Printable String and a Teletex String syntax. There are rules about mixing syntaxes in values for these attributes.

The next two sections explain what you must do if you want to include Domain-defined attributes (DDAs), Organisational Units. and Personal Name attributes in an object of the OR Address class.

HP recommends that you should always supply a Printable String value for these attributes (in addition to a Teletex string value, if you need that), in case you need to interwork with a system that does not support the Teletex String syntax.

Mixing Syntaxes in Organisational Units and DDAs

An application should treat Organisational Units and DDAs as lists. Therefore if you use a particular syntax for one of these attributes within the list, you must also use that syntax for any previous member of the list.

For example, the Organisational Unit attributes in this list have valid values:

OU1: Teletex, Printable
OU2: Teletex, Printable
OU3: Printable

However, for ease of maintenance and of interworking, it is advisable to use consistent values for all members of the list (in this case, OU3 would share the possible syntaxes of OU1 and OU2).

For DDAs, there is a further consideration: DDAs have a type and a value. For each DDA in the list, the type and the value must have matching syntaxes. So rules governing the matching of syntaxes affect both the list itself and each individual DDA in the list:

- A member of the list must have a syntax that matches that of the previous member of the list — for example, if Domain Type 2 has a Teletex String syntax, Domain Type 1 must also have a Teletex String syntax.
- For any DDA, the syntax for both type and value must match — for example, if Domain Type 1 has a Teletex String syntax, Domain Value 1 must also have a Teletex String syntax.

Personal Name Attributes

The Personal Name attributes are Surname, Given Name, Initials, and Generation. When dealing with Personal Name attributes, an application should take account of the following: the Surname within the Personal Name attributes is mandatory, and the Personal Name attributes are considered as a group. Therefore, if you provide a value with one syntax for a Surname attribute, you must supply values with the same syntax for any other Personal Name attributes. For example, if you supply a Teletex String value for Surname, you must also supply Teletex String values for any other Personal Name attributes you use. Similarly, if you supply a Printable String value for Surname, you must also supply Printable String values for any other Personal Name attributes you use.

Value Positions for Multi-Valued Attributes

In the MAILbus 400 API, attributes of an object of the OR Address class that have both a Printable String and a Teletex String syntax are multi-valued attributes: the Printable String goes in value position `0`, and the Teletex String goes in value position `1`.

The MAILbus 400 API will accept a Teletex String value with no Printable String equivalent for these objects, and may present your application with one. Note, however, that the Teletex String value will still be in value position `1`; value position `0` will not have a value.

Part IV

Background to Previous Kits

This part gives a brief description of Version 3.2-12 of the MAILbus 400 MTA and MAILbus 400 API. This information was available in the Release Notes for the previous kits and so might be familiar to you. It includes:

- Documentation Provided (see Section 9).
- Interworking information (see Section 10).

9 Documentation Provided

The following sections list the documentation provided with the MAILbus 400 MTA and MAILbus 400 API.

9.1 With the MAILbus 400 MTA

The documentation provided with the MAILbus 400 MTA is written for the MAILbus 400 MTA running on the OpenVMS or Tru64 UNIX® operating systems. Information provided in the documents listed below applies to both operating systems, unless specifically indicated:

- *MAILbus 400 Getting Started*, Version 3.2-12

  This guide is new to the MAILbus 400 MTA documentation set and describes how to get your messaging system, based on the MAILbus 400 MTA and its products, started quickly.

- *HP MAILbus 400 MTA Planning and Setup*, Version 3.2-12

  This guide contains information about how to plan and set up a Message Transfer System (MTS) based on the MAILbus 400 MTA. This guide also contains some introductory information and the MAILbus 400 MTA Glossary.

- For OpenVMS Alpha, *HP MAILbus 400 MTA Installing on an OpenVMS Alpha System*, Version 2.0
  For OpenVMS VAX, *HP MAILbus 400 MTA Installing on an OpenVMS VAX System*, Version 2.0

  These installation cards describe how to install the MAILbus 400 MTA on a OpenVMS Alpha and OpenVMS VAX system, respectively. Use this card in conjunction with Part III of *HP MAILbus 400 MTA Planning and Setup* and these Release Notes.

- *HP MAILbus 400 MTA Tuning and Problem Solving*, Version 3.2-12

  This guide explains how the MAILbus 400 MTA works, how to modify MAILbus 400 MTA attributes according to your management requirements, and how to solve problems that might occur in an MTS based on the MAILbus 400 MTA.

In addition to these guides, reference information is available in the *MTA Module Online Help* and *MTS Module Online Help*.

9.2 With the MAILbus 400 API

The documentation provided with the MAILbus 400 API is written for the MAILbus 400 API running on the OpenVMS and the Tru64 UNIX operating systems. Information provided in the documents listed below applies to both operating systems, unless specifically indicated:

- For OpenVMS Alpha, *MAILbus 400 API Installing on an OpenVMS Alpha System*, Version 2.0
  For OpenVMS VAX, *MAILbus 400 API Installing on an OpenVMS VAX System*, Version 2.0

  These installation cards describe how to install the MAILbus 400 API on an OpenVMS Alpha and OpenVMS VAX system, respectively.

- *MAILbus 400 API Programming*, Version 1.4

  This guide describes all the packages, functions and data types provided by the MAILbus 400 API.

  This guide has not been updated for MAILbus 400 API Version 2.0. Any documentation amendments are described in Section 8.

10 Interworking

The following subsection provides information about versions of related products that you need to be aware of if you are using these products with the MAILbus 400 MTA.

10.1 Interworking with Microsoft Exchange Server

The Microsoft Exchange Server™ is a mail application that generates File Transfer Bodyparts. If your mail system includes the Microsoft Exchange Server, you will need to set the Content Type on the O/R address entry, or entries, for the Microsoft Exchange users as 1992 File Transfer IPMS.

HP MAILbus 400 MTA Tuning and Problem Solving describes the object identifiers for the Content Types, and in particular the File Transfer Bodypart, in more detail.

Be aware that if your mail system also includes ALL-IN-1™ Version 3.2 or LinkWorks, an interoperability problem has been identified. This interoperability problem prevents the Microsoft Exchange Server receiving any message from either ALL-IN-1 Version 3.2 or LinkWorks over a 1992 connection from the MAILbus 400 MTA.

Microsoft will be providing a fix for this interoperability problem, which will be available by the end of April 1996, and on request. To obtain this fix, contact the support center for your Microsoft Exchange Server.

10.2 Interworking with the VAX Message Router X.400 Gateway

If you intend to use the MAILbus 400 MTA with the VAX Message Router X.400 Gateway (MRX), make sure that you have installed MRX Version 2.3 or later. To obtain MRX Version 2.3 contact your HP support center.

When using the MAILbus 400 MTA with MRX you must ensure that the MRX MTA control flag GOSIP_CONFORM is set for each peer MTA record relating to a MAILbus 400 MTA.

Note that MRX acts as an MTA conforming to the 1984 messaging standards.