A White Paper On

Externally Defined Body Parts (Body Part 15): Issues and Recommendations

Version 1.3

Prepared for

The Electronic Mail Association PRMD Operators' Committee

Prepared by

Mohsen Banan

Tele-Voice

17005 SE 31st Place

Bellevue, WA 98008

April 14, 1993

Contents

Executive Summary

Today many messaging systems can transfer application data without regard to its type but are not capable of automated identification of the information contained in messages on a global or multi-vendor basis. Often, the originator of the message has to explicitly specify (e.g., in the subject line) the type of application data (e.g., spread-sheet) and the recipient has to take explicit actions to process the information when the message is received. Automated identification of application data types by messaging systems enables capabilities including automatic application invocation, application data type conversion, and, at the very least, enables the recipient to know the type of data received. The best e-mail systems are those that automate and facilitate secure processing of application data to the point that all of the complexities of the procedure are hidden from the user.

The 1988 CCITT X.400 Recommendations include a definition for an Externally Defined Body Part (Body Part 15), allowing the specific identification of various types of information to be exchanged between User Agents (UA). X.400 user agent and message transfer agent (MTA) implementations providing the full benefits of application integrated messaging are expected in the near term. Some examples of the benefits of application integrated messaging include the exchange of revisable files, mail-enabled applications, electronic form routing and approval systems, and mail-aware group scheduling systems.

When transferring data through messaging systems, the ability to automatically identify body part types within a message, and to have that type recognizable to a receiving application, greatly increases the utility of the information to its recipient. In addition to the benefits of application integrated messaging, the ability to automatically identify a body part type may increase the level of transparent interpretability between mail senders and receivers, potentially of any platform, by allowing automatic conversion to a recipient's preferred formats.

For example, a sender can create a document using word processor A. This document can then be automatically packaged as an instance of a body part type defined for word processor A and mailed to a receiver on a different platform who wishes to receive it in a format acceptable to word processor B. The recognition of the word processor type and the conversion between formats could be handled by either the MTA or the UA, so that all processing is transparent to the user.

In order to realize the business benefits of transparent interoperability, where information processed in one user's environment can be properly transferred to another user's environment for further processing, many infra-structural elements need to be in place. These include global identification and recognition of application data which is a critical element in the evolution towards transparent interoperability.

In order for a data identification method to be useful on an international messaging level, there must be recognized and published, globally unique object identifiers for data types. Currently, there is no widely adopted method for vendors to register and publicize their application data types and identifiers. Before the benefits of application integrated messaging can be realized, document and application vendors must first register object identifiers for each of their popular data types.

EMA's eventual goal is to provide application, document and messaging vendors with consensus user input and guidance regarding how to implement and support object identification and encoding, to the benefit of both vendors and users alike.

In this White Paper the PRMD Operators' Committee (POC) recommends that document creation vendors register with a registration authority and receive a unique identifier to be used in assignment of Object Identifiers representing their document formats. This will be the first step in enabling the global exchange of fully formatted business documents. In addition, the POC recognizes that there needs to be an authorized publisher of this list of Object Identifiers. This paper also discusses the scope and responsibilities of the publisher.

Once processes and procedures for identification of application data on a global basis are in place, widespread usage of mail-enabled/mail-aware applications will follow and the full benefits of application integrated messaging can be realized. Interoperability of applications and automated processing of application data through messaging systems results in greater work group productivity.

Chapter 1

Introduction

The inclusion of application data (e.g., spread-sheets) in the transmission of electronic messages is becoming increasingly common. Although most messaging systems provide the ability to transfer data without regard to its type, few messaging systems are capable of identifying the information contained in messages on a global or multi-vendor basis. Often, the originator of the message has to explicitly specify (e.g., in the subject line) the type of application data (e.g., spread-sheet) and each recipient has to take the responsibility for recognizing the data type, determining if any conversion or processing is needed, and then invoke the converter or other applications as needed. All of this manual processing should be automated by the messaging system.

As the interconnection of electronic mail systems becomes common place, simply providing messaging interconnectivity is no longer viewed as sufficient. Users of an electronic mail service expect a high level of overall service and functionality from their messaging system. A large part of this overall service is the ability to send and receive all types of documents, without concern to the preferences and limitations of one's messaging partner.

Considering the Electronic Mail Association's commitment to X.400, – the emerging international standard messaging system – as the basis for implementation of a global messaging environment, this paper focuses on the transfer of application data through X.400 messaging systems. In X.400 specifications, the Externally Defined Body Part (Body Part 15) is one of the means of transferring application data.

1.1 About Externally Defined Body Part

The ability to identify, in a standard and globally unique way, types of enclosed application data is desirable to increase the utility of the application data to its recipients. Identification of application data types by messaging systems enables capabilities including automatic application invocation and application data type conversion. At the very least, it enables the recipient to easily know what type and encoding of data is being received, and without manual intervention by the originator.

In X.400, Externally Defined Body Parts (Body Part 15) may be used to facilitate transfer of application data types. The Externally Defined Body Part includes a "label" which travels with the message and identifies part of the whole message as being of a particular type, a spread-sheet, for instance. The Externally Defined Body Part represents an information object whose semantics and format are denoted by an Object Identifier (OID) that the body part carries.

On the basis of externally defined body part type, all body part types are divided into two classes, *basic* and *extended*. The basic body part is a body part other than externally defined such as text, voice or facsimile. Throughout this paper we refer to Body Part 15, Externally Defined Body Part and Extended Body Part interchangeably.

Currently, a technical mechanism exists for identifying application data types in X.400 messages. However, there are no widely adopted means of obtaining unique identifiers for actual application data types and subsequently using these identifiers in messages. This paper identifies the issues surrounding the standardized identification of application data types.

1.2 Purpose and Scope

The Electronic Mail Association's PRMD Operators' Committee has identified the following goals and strategy surrounding Body Part 15 registration issues:

```
POC Goal:
       ``Enable techniques to support enhanced application
       recognition of computer data for both intra- and
       inter-company data exchange.''

POC Strategy:
       ``By educating Application, Document and Messaging
       Vendors on the importance of common, global Object
       Registration, get vendors to register with a national
       registration body (such as ANSI), and assign globally
       unique Object Identifiers (OIDs) across all document
```

`types.''`

Based on this, the PRMD Operators' Committee has determined that an educational paper discussing the registration of object identifiers and the representation of object data, as it relates to electronic mail, would be of value to the members of the Electronic Mail Association as a whole. This paper is prepared for that purpose.

In order to fulfill the user requirement of transparent interoperability through messaging systems, many infra-structural elements need to be in place. Global identification of externally defined body part types, their parameters and their encoding of application data is a critical element in the evolution towards transparent interoperability. The purpose of this paper is to identify and analyze issues surrounding the transfer of application data through messaging systems.

This paper identifies techniques to support enhanced application recognition of computer data for both intra- and inter-enterprise messaging exchange. This is accomplished by defining application data transfer in messaging systems, exploring the issues and problems relating to the transfer and automated processing of application data through messaging systems, and then by recommending solutions to the problems identified.

This paper serves as a tool in educating application, document and messaging vendors and users on the importance of common and global Object Registration. It is hoped that through awareness of the Body Part 15 related issues and the benefits that can be gained from realization of it, vendors will be motivated to register with a national registration body (e.g., ANSI) and to assign globally unique Object Identifiers (OIDs) across all document types.[1] Ultimately, a goal is to get application, document and messaging vendors to provide recognition of OIDs within their products.

Another explicit objective of this paper is to collect in one place the results of the efforts that the POC has invested in investigating Body Part 15 related issues.

Note: Issues relating to content types and body types that are explicitly recognized in the X.400 specifications (e.g., EDI) are outside the scope of this paper. Other efforts addressing application data and document type identification (e.g., OMG[2], ODA) are complimentary and not in conflict with the intended goal of this document. Here, the focus is on the integration of application data into X.400 messaging systems.

[1] It is possible to register OIDs for Extended Body Part Types with registration authorities other than ANSI.

[2] Object Management Group (OMG) is a vendor and user consortium defining an object oriented architecture for the industry.

1.3 Intended Audience

The main audiences of this paper are application, document and messaging vendors wishing to integrate application data into messaging systems and the middle level management staff operating a PRMD. PRMD Operators' management staff, PRMD Operators' technical staff, messaging system architects and those wishing to understand the issues surrounding the transfer of application data through messaging systems can all benefit from this paper. The structure of this paper is intended to allow for all of these audiences.

Those unfamiliar with X.400 will find the tutorial chapters useful. In fact, some readers may wish to first read the tutorial chapters. These tutorials specifically cover topics directly related to Body Part 15 and are quite brief. Readers who are interested in more detailed explanations of messaging concepts should refer to other messaging tutorial publications. A glossary of relevant terms and a list of abbreviations used throughout the paper are included as appendices.

It is assumed that the reader has some knowledge of X.400 Message Handling Systems, and the practical aspects of deploying mail user agents (UAs) and gateways. Where necessary, the paper provides technical information (although generally not in great detail) and therefore should prove valuable to technical PRMD Operator staff as well as to the managers responsible for realizing the transfer of application data.

It is intended that the PRMD Operators' Committee will present this paper to the Electronic Mail Association. Members representing public mail carriers, private companies, government entities, academic institutions, and those in the standards communities will find the issues discussed in this paper and the recommended solutions to identified problems to be of immediate importance to them in facilitating the transfer of application specific data formats by electronic messaging systems.

1.4 About this paper

This section includes information about the structure of this paper and also describes how different readers can benefit from reading specific sections of this paper.

The structure of this paper is as follows.

Chapter 2 (see section 2 [Messaging Transfer of Application Data], page 9), provides an overview of key concepts relating to the transfer of application data through messaging systems.

Chapter 3 (see section 3 [Relationship to Existing Systems], page 25), considers the relationship of 1988 X.400 Externally Defined Body Part mechanism with other existing similar systems.

Chapter 4 (see section 4 [Issues], page 33), focuses on issues and problems attendant to messaging transfer of application data such as the need for vendors to register application data types.

Chapter 5 (see section 5 [Recommendations], page 37), identifies the steps that can be taken to facilitate the widespread use of externally defined body part types, their parameters and their encoding of application data types in messaging systems.

The appendices include tutorial and related information that make this document more self contained.

The structure of this paper is intended to allow its effective use by readers with varying degree of familiarity to this paper's topics.

Those already familiar with the basic issues surrounding the transfer of application data in X.400 messaging systems may wish to focus on Chapters 3, 4 and 5.

Those unfamiliar with the basic issues surrounding the transfer of application data in X.400 messaging systems can best benefit from this paper by first reading the tutorial sections and then reading chapters 2, 3, 4 and 5. The glossary and abbreviations appendices can be helpful to those unfamiliar with the X.400 terminology.

The concept index in the back of the paper can be used to locate information about specific topics. It makes it easy to use this document for reference purposes.

Chapter 2

Messaging Transfer of Application Data

Standardization within the field of Information Technology is resulting in standardized information objects that must be unambiguously identifiable on a global basis. Standards organizations, government entities and private companies may define information objects which may or may not be OSI objects. Such information objects may represent a variety of real objects, e.g., application data, document types, etc. This chapter focuses on topics related to the integration of information objects into X.400 messaging systems.

The ability to identify, in a standard and globally unique way, types of enclosed application data is desirable to increase the utility of the application data to its recipients and also to facilitate implementation of mail-enabled and mail-aware applications. For example, a meeting scheduler might define its application data as a standard representation for information about proposed meeting dates. An intelligent user agent would use this information to conduct a dialog with the user, and might then send further mail based on that dialog.

2.1 About Application Integrated Messaging

In the messaging model, the ‘‘User Agent” (UA) is a generalized concept, not limited to traditional e-mail application programs used to compose messages. The user agent can be any application that knows how to interact with the Message Transfer System (MTS). For example, if you wanted to send someone a spreadsheet document, the spreadsheet application could be ‘‘mail-enabled” and use the Message Transfer System to send your spreadsheet to its intended recipients. On the receiving end, the user agent can recognize the format of the application data (in this example, spreadsheet) and invoke the appropriate spreadsheet application. The scope and extent of mail-enabled applications are only limited by the user/developer's imagination.

In order to transparently include arbitrary application data within an e-mail message, and also to have that data successfully translated back to its original form upon delivery of the e-mail message, there must be some way of uniquely identifying the included data.

In X.400 specifications, the externally defined body part – Body Part 15 – is the mechanism for inclusion of application data, and Object Identifiers are used to uniquely identify the type and encoding of application data. See section A [Tutorial on Messaging Transfer of Application Data], page 43.

The benefits of integration of application data into messaging systems can be successfully manifested if document and application vendors define X.400 extended body parts for their word processor documents, spreadsheets, images, etc. Once object identifier values for application data types are in place, messaging user agents can recognize the application data contained in messages and "launch" the desired application to process the information object received in messages.

Automated processing of application data

UA implementations may provide different mechanisms for selecting what application should be launched upon receipt of the particular application data contained in a message. This mechanism may be fixed (static) or extensible (dynamic). UA implementations offering mechanisms that allow for dynamic – "on the fly" – configuration of a UA's table of known extended body part types provide more flexibility to PRMD operators.

Figure 2.1 diagrams an example of a UA implementation capable of dynamic recognition and processing of extended body part types. In this example, the "UA's Extended Body Part Table" may consist of an extensible number of entries, each of which associate a body part type – identified by an Object Identifier – with a specific application program that can process the information object contained in the body part. After receiving a message from the MTS, the UA consults its table and decides what application to invoke to process the extended body part contained in the message.

From a theoretical perspective, using the Body Part 15 feature, UAs can automate the invocation of any application on the application data contained in messages. While this can provide enormous benefits, it is also important to recognize the serious security implications of this capability.

Figure 2.1: Automated processing of Application Data

Security Considerations

To illustrate the security threats that the above mentioned capabilities pose, we consider an obvious example. Consider the operating system shell script language (e.g., BAT files for DOS) as an extended body part. If a UA implementation was to recognize the "shell script" extended body part, launch a "shell" and give it the shell script contained in the message, the results could possibly be very undesirable, issuing for instance, a command to "delete all files".

Although it is obvious that the above example is open to substantial abuse and it is unlikely that such scenarios will be permitted, the security threat of executing programs on data coming over the network should be taken very seriously by all PRMD Operators. This does not just relate to the implementation of Body Part 15 technology, but to any inter-company data exchange which may involve automated processing of application data as well.

The following is a non-exhaustive list of security related issues to consider:

- Many application programs include mechanisms that provide access to the operating system shell or execution of other programs. Such facilities are quite obviously open to substantial abuse. Once that door is open, the ramifications are very serious. Therefore, unrestricted execution of programs that may execute other programs should not be permitted.

- Some applications include dangerous features that can be exploited. For example, the execution of general-purpose Postscript interpreters entails serious security risks, and PRMD Operators are discouraged from simply sending Postscript e-mail body parts to any "off-the-shelf" interpreters. While it is usually relatively safe to send Postscript to a printer, where the potential for harm is greatly constrained, PRMD Operators should consider many factors[1] before they add interactive display of Postscript bodies to their mail readers. Dangerous operations in the Postscript language include, but may not be limited to, the Postscript operators `deletefile`, `renamefile`, `filenameforall`, and `file`. Other applications may also include features that are open to abuse. The security risks of integrating each specific application into the messaging system should be considered before permitting the processing of any body parts corresponding to its application data types.

- Known bugs may be exploited. Apart from noting this possibility, there are no specific actions that can be taken to prevent this, apart from the timely

[1] The MIME RFC, [?], enumerates some, though probably not all, of the possible problems with sending Postscript through the e-mail.

correction of such bugs if any are found.

- Security attacks that take advantage of the messaging system in order to propagate themselves are especially serious. Application data containing a ‘‘worm” is an example of such an attack. A worm is a program that replicates itself and spreads, but does not attach itself to other programs. Application data containing a ‘‘virus” is another example of a security attack that can come through the messaging system. A computer virus infects a system by attaching itself to other programs and converting them into viruses. A worm differs from a virus in that it does not require another program in order to survive. While viruses spread by sharing infected software, worms usually spread over a network. Worms can potentially paralyze a network, as did the famous Fall 1988 Internet worm. Therefore, special attention should be given to processing of application data that can result in re-use of the messaging system.

PRMD Operators can avoid many of the potential problems mentioned above by careful selection of the application data types that are integrated into the messaging environment.

2.2 Benefits of Application Integrated Messaging

The best e-mail systems are those that automate and facilitate secure processing of application data to the point that all of the complexities of the procedure are hidden from the user.[2] The Body Part 15 mechanism provides for realization of such e-mail systems. X.400 user agent implementations providing the full benefits of application integrated messaging are expected in the near term.[3]

Briefly, the benefits of application integrated messaging include:

- **Exchange of Revisable Files** – word processing, graphics, data base and spreadsheet applications can travel through the messaging system eliminating the need for re-keying or reformatting, hence speeding up collaborative editing cycles, and boosting the quality and efficiency of office output. Further, when conversion between file types is possible, collaborative editing can be done in each user's preferred format.

[2] Readers interested in learning more about basic e-mail workflow benefits on an industry by industry basis are encouraged to consult the EMA's ‘‘The Electronic Mail Advantage”.

[3] ‘‘EMA's 1988 X.400 Migration” white paper details many of these benefits. Some of the benefits are reproduced in this section.

- **Electronic Form Routing and Approval (EFRA)** – ERFA technology can be used to automate many business activities such as purchase requisitions or project hours reporting.

- **Group Scheduling Systems** – The ability to send cross-platform e-mail messages from within a scheduling package is a significant benefit that is relatively simple to implement. On the receiving end, a scheduling notice that can be automatically identified and processed by the user agent greatly increases the benefits of the messaging system and the scheduling system to the user. The non-real time nature of e-mail limits the scope of group scheduling operations which can be reliably implemented based on messaging services. For example, transaction oriented scheduling operations, such as finding free calendar slots and reserving those time slots, in heterogeneous environments can not be reliably implemented based on messaging services.

- **Mail-Enabled Applications** – Interactions between users and applications, applications and data bases, and users and data bases through the messaging system are some examples of mail-enabled applications. While technology is necessary, it is how an application applies the technology which is really important to companies. It is through mail-enabled applications that most benefits of messaging technology will be realized for the user. The scope of mail-enabled applications, performing actions transparent to users by recognizing OIDs, is virtually unlimited.

From the user's perspective, the added functionality that application integrated messaging offers directly results in improved productivity. From the vendor's perspective the benefits of application integrated messaging translates into larger demand and larger customer base.

It is important to note that these benefits can only be realized after document and application vendors register the object identifiers corresponding to their information object types.

2.3 About Transfer of Application Data in X.400

The 1984 X.400 Recommendations defined several body part types, including IA5 text (used for plain text), teletex data, voice and group 3 facsimile data. Shortly after initial publication, the Bilaterally Defined Body Part (also known as Body Part 14 and Unidentified Body Part) was added as an option. The Body Part 14 mechanism makes it possible to include arbitrary types of data objects within X.400 messages.

However, it does not provide for identification of the enclosed data in a standardized way. As the name "Bilaterally Defined" implies, this body part type can be used for any form of data transfer and identification that the originator–recipient pair have agreed upon. In the absence of pre-established agreements for identification of application data in Body Part 14, often, the originator of the message has to explicitly specify (e.g., in the subject field) the type of application data being sent. The recipient then must take explicit action to process the information received in the message. It is important to note that Body Part 14 is being extensively used in today's messaging environment and is supported by most vendors providing X.400 products.

The shortcoming of the bilaterally defined body part is that different arranged uses for it may interfere with each other.

The 1988 X.400 Recommendations solve many of the limitations of Body Part 14 with the introduction of the externally defined body part type. The externally defined body part type (also known as Body Part 15 and Extended Body Part Type) enables vendors, or a community of users to define its own body part type for any specific purpose. This is accomplished by assigning a unique identifier to the body part type.

The requirement of standardized file transfer through messaging systems is addressed in the 1992 X.400 Recommendations. In the 1992 X.400 Recommendations the file transfer body part type is defined as an externally defined body part type.

2.3.1 1988 Implementations

The `ExternallyDefinedBodyPart` was first introduced in the 1988 X.400 recommendations. The 1984 recommendations did not include provisions for the ExternallyDefinedBodyPart.

Not all of the 1988 features need to be implemented for users to benefit from the Body Part 15 facility. Specifically, only support for Body Part 15 in the protocol implementation and the UA capability registration[4] features are needed. Other 1988 features such as Use of X.500, Distribution Lists, Message Store, Physical Delivery, Security, ... need not be implemented for the Body Part 15 benefits to be realized.[5]

The MHS Special Interest Group (SIG) section of the Open Systems Environment Implementors' Workshop (OIW) Implementor Agreements [?], makes a number of recommendations for using the externally defined body part. This paper is consistent with those agreements.

[4]The UA capability registration feature provides for a UA to register its capabilities (e.g., the ability to accept and process certain Body Part 15 information objects) with the MTA.

[5]Refer to "EMA's 1988 X.400 Migration" white paper for more details.

The 1988 X.420 [?], introduced the EXTENDED-BODY-PART-TYPE definition which facilitates the incorporation of body parts not specifically defined in the 1988 X.400 recommendations. By taking advantage of this feature, it is possible to benefit from EXTENDED-BODY-PART-TYPEs defined in the 1992 X.400 Recommendations. The file-transfer-body-part is an example of such a scenario. See section 2.3.2 [1992 File Transfer Body Part], page 16, for more information.

2.3.2 1992 File Transfer Body Part

The 1992 X.400 recommendations introduce a new extended body part type that has been designed specifically to meet the users' requirements for file transfer.

A file-transfer-body-part represents an information object used to convey the contents, and optionally, the attributes of a stored file. The file-transfer-body-part is based on the file model defined in FTAM.[6]

Since file-transfer-body-part is an ExternallyDefinedBodyPart type (tag 15), it is possible to benefit from its definition in 1988 implementations.

The file-transfer-body-part allows for transferring single or multiple documents of various types. In addition to conveying the specifics of the format that is contained in these body parts, file related information may also be included. To fulfill the requirement of transferring files using the messaging system, the file-transfer-body-part may be the most suitable method.

The advantage of using file transfer body part type is that, in a unified way, it accommodates inclusion of additional file related information within the extended body part. A brief description of some of the file transfer parameters follows.

related-stored-file parameter indicates any intended relationship between the file contained in this message and any file(s) held by the recipient or a prior message.

content-type parameter indicates the type and structure of the information constituting the file contents. This is specified using an object identifier.

environment parameter describes the environment (e.g., machine, operating system, processor and version of application) in which the file originated. It also indicates how the file is to be treated upon receipt.

compression parameter indicates if the file contents have had a compression algorithm applied. The compression algorithm is also specified by an object identifier.

[6] The ISO standard for File Transfer.

file-attributes parameter conveys values of any of a set of optional file attributes. The file attributes are aligned with FTAM. The file attributes include pathname, permitted actions, date and time, object size, access control, ...

As of this writing, the working text[7] of the OIW MHS SIG includes recommendations about the use of the file transfer body part type. The following segment is reproduced from the June 1992 working text of the OIW.

> D.6.4 Use of File Transfer Body Part Type
>
> The File Transfer body part type is the recommended mechanism for the exchange of complex computer data via intra- and inter-company X.400 messages. It enables automatic type recognition for the file being sent and, possibly, automatic invocation of the appropriate application necessary to process the data.

The OIW MHS SIG working documents provide additional information on the use of file transfer body part type. Those interested in this subject should monitor the evolution of these agreements. March 1992 OIW stable agreements do not mention the use of file transfer body part type.

It is important to note that the `file-transfer-body-part` is a compatible extension to the Body Part 15 mechanism. Registration and publication of object identifiers for application data are essential to the definition of both file transfer body parts and extended body parts.

The decision on whether to use the `file-transfer-body-part` or the "plain" extended body part type is one that can best be made based on the specifics of the application data to be transfered. The decision also needs to take into consideration about interworking with other systems in terms of availability of this feature and of the specific purpose of file transfer parameters that are included (e.g., what can the application do with this information).

The advantage of using file transfer body part type is that in a unified way `file-transfer-body-part` accommodates inclusion of additional file related information in an extended body part type. The current OIW working texts suggests that the Encoded Information Type (EIT) of messages containing FTAM body parts do not reflect the details of the FTAM body part content. This would limit the ability of MTAs to detect the need for body part conversions. The use of a new extended body part type, with its own EIT, would allow MTAs to recognize the need for conversions, leading to conversion or non delivery, depending on the capability of the MTAs. The CCITT MHS Implementors' Guide (Version 9) allows optional inclusion of OID in the EIT to reflect the application data type of the File Transfer Body Part.

[7] Not the OIW Stable Implementor Agreements.

In some cases it may be desirable to define the file transfer body part type parameters for a particular application data in addition to defining an extended body part type for it. By having both types defined, based on the context of the communication the user can decide which option is better suited for a particular use.

The process of defining application data for inclusion in file transfer body parts is in many ways similar to the process of defining extended body parts which is described in the next section.

2.4 Defining an Extended Body Part Type

In X.400 messaging systems, integration of application data into the messaging system can be accomplished by defining an extended body part type for the application data.[8] However, before creating a new extended body part type, the application developer should carefully consider the use of the existing standardized framework of the File Transfer Body Part for the application data.

The definition of the extended body part must include the *complete* `EXTENDED-BODY-PART-TYPE` definition (see section A.3.1 [Externally defined body part], page 47). An example of the definition of an extended body part is included in this section.

Once the definition of the `EXTENDED-BODY-PART-TYPE` has been completed, Object Identifier assignments should be made.

About Object Identifiers

An Object Identifier (OID) is a value (distinguished from all other such values) which is associated with an information object. When properly assigned an Object Identifier is globally unique and can readily be processed without the need to know where it comes from.

However, today there is no global register of what the OIDs map to – their specific meaning. See section 5.2 [Publishing Extended Body Part OIDs], page 39, for more details about this issue.

A brief tutorial on Object Identifiers is included in this paper. See section B [Tutorial on Object Identifiers], page 51.

In order to guarantee that an Object Identifier is unique and unambiguous, it must be registered with a Registration Authority. This paper focuses on the specifics

[8]Throughout this paper we refer to Extended Body Part, Externally Defined Body Part and Body Part 15 interchangeably.

relating to Registration of OIDs for Body Part 15.

Obtaining Object Identifiers

Although Body Part 15 object identifiers must be globally unique, not all Body Part 15 OIDs have to be assigned by the same registration authority. In fact, object identifiers for Body Part 15 will be assigned by many organizations, each of which has the choice of selecting its position in the object identifier tree.

There are many ways to obtain object identifiers. One such way is described below:

1. The application provider obtains a unique Numeric Name for their organization from a registration authority such as ANSI in the US. Relevant information for contacting the ANSI Registration Authority is included as an appendix to this document. See section E [ANSI Registration Authority Contact], page 67. Relevant information for obtaining and using a unique Numeric Name is contained in ANSI ISSB 840, ISSB 843 and ISSB 989. Once a Numeric Name has been obtained from ANSI the registering organization uses this Numeric Name in order to form an object identifier denoting their organization. An example is provided later in this section.

2. The application provider (organization) allocates a series of numbers to identify the application data format; these numbers are appended to the object identifier constructed in step (1) to form an object identifier that is globally unique. In general, it is recommended that the application provider (organization) use a hierarchical structure for identifying their data types to ease in the administration of the identifiers. An example is provided later in this section.

Figure 2.2 illustrates an example OID tree. In this figure, the dashed lines represent arcs that can be root hierarchies for other object identifiers. It is possible for Body Part 15 OIDs to fall under any of the dashed lines. Organizations wishing to define a Body Part 15 OID may be able to obtain an arc below any of the dashed lines.[9] Each circle represents a Registration Authority. The circles containing `X.208` also mention the relevant section of X.208 document where more pertinent information can be found. For example, (B.1) refers to X.208, Annex B, paragraph 1.

The `document ''Y''` example which this OID tree leads to is described next.

[9] Not necessarily immediately below.

2.4.1 Example Definition of an Extended Body Part Type

In order to illustrate what is required for an application/document vendor to define an extended body part type so that the vendor's application data can be integrated into X.400 messaging systems, an example is used.

Consider as an example, "ABC" vendor who wishes to define an extended body part type for document "Y".

Every extended body part type must be defined by means of:

1. `OBJECT IDENTIFIER`s (`OID`s) that uniquely and unambiguously identify the body part.

2. The ASN.1 `EXTENDED-BODY-PART-TYPE` macro.[10] See section A.3.1 [Externally defined body part], page 47, for the definition of `EXTENDED-BODY-PART-TYPE`.

In the following sections we show how "ABC" vendor satisfies these requirements.

Obtaining OIDs

After registering its name form and obtaining a unique number form with ANSI,[11] "ABC" vendor defines a root OID for its organization. For the purposes of this example, let us say that "ABC" vendor had obtained the unique Numeric Name "n" from ANSI. ABC vendor can define the root OID for its organization as:

```
abc-vendor OBJECT IDENTIFIER ::=
               { joint-iso-ccitt(2) country(16) us(840)
                          organization(1) abc-vendor(n) }
```

Until recently the OID assigned to the organization under ANSI had the form of:

```
abc-vendor OBJECT IDENTIFIER ::=
               { iso(1) member-body(2) us(840) abc-vendor(n) }
```

[10] The 1988 X.420 [?] dictates the use of `EXTENDED-BODY-PART-TYPE` macro.

[11] In this example, "ABC" vendor has chosen to be under ANSI in the OID tree. Body Part 15 OIDs can be registered under other branches as well.

Both the *old* and the *new* locations in the OID tree are illustrated in Figure 2.2. The "move" to the new location in the OID tree is a recent event. Prior to registering with ANSI each organization should contact ANSI to obtain the most recent information about their policies and procedures. Relevant information for contacting the ANSI Registration Authority is included as an appendix to this document.See section E [ANSI Registration Authority Contact], page 67.

Based on the root OID for its organization (`abc-vendor`), "ABC" vendor now has its own Object Identifier name space and can generate Object Identifiers as needed. In order to manage this name space, "ABC" vendor has created an internal registration authority that assigns OIDs under the `abc-vendor` arc. ABC vendor uses a hierarchical structure for identifying their data types to ease the administration of the identifiers. One of these registration hierarchies is for MHS body parts. ABC vendor may define OIDs for information objects other than body parts, and those can go under a different registration hierarchy (arc).

For example, "ABC" vendor can define *mhs-body-parts* root OID as:

```
abc-vendor-mhs-body-parts OBJECT IDENTIFIER ::= {abc-vendor  1}
```

The purpose of this definition is to gather all extended body part OID definitions in one place.

Defining the EXTENDED-BODY-PART-TYPE

In order to keep this example simple, let us assume that document "Y" format requires no parameters. The following Extended Body Part is defined for document "Y" data format.[12]

```
document-y-body-part EXTENDED-BODY-PART-TYPE
         DATA            OCTET STRING
         ::= document-y-id-data

document-y-id-data OBJECT IDENTIFIER ::=
          { abc-vendor-mhs-body-parts 1 }
```

[12]Example definitions which include parameters can be found in X.420 (1988), Annex B.

Since document “Y” does not follow ASN.1 encoding rules and is octet aligned, the encoding choice for `Data External` is `octet-aligned`. In this example, `DATA` is defined as `OCTET STRING` which indicates that the syntax of document “Y” is `octet-aligned`. The specific syntax for document “Y” is described through an authorative reference to a specification in which “ABC” vendor defines document “Y”.

Once the extended body part type definition for document “Y” is complete, the X.400 messaging system can recognize messages containing extended body parts tagged with the object identifier:

```
{ (2) (16) (840) (1) (n) (1) (1) }
```

as ones having the syntax of document “Y”. Based on this, the UA may start processing the “Y” document automatically.

Publishing the newly defined extended body part type

Once an extended body part has been defined, its integration into messaging systems is possible. However, the specification of the newly defined body part must be made known to application developers, messaging implementors and users before it can be useful. A Body Part 15 publishing entity can be used to publicize the specification of the newly defined body part types on a global basis. At this time, such a publishing entity does not exist. This issue is dealt with in some detail later in this paper (see section 4.2 [Body Part 15 OID Publication and Registration], page 34). Recommendations about initial steps that can be taken to address this issue are also dealt with in this paper. See section 5.2 [Publishing Extended Body Part OIDs], page 39, for details of these recommendations.

Chapter 3

Relationship to Existing Systems

This chapter considers the relationship of Body Part 15 methods of inclusion of application data into X.400 messaging systems, with other existing messaging systems' method of transferring application data. The interoperability of X.400 messaging systems that support externally defined body parts with other existing messaging environments is very desirable. Accommodating the richest possible level of interoperability among these systems is a goal of this paper. The emphasis here is to allow an X.400 user to exchange arbitrary information objects with users on other existing systems as transparently as possible.

Many existing messaging systems can transfer application data without regard to its type but are not capable of identifying the information on a global or multi-vendor basis. Body Part 15 capability of identifying different types of application data in a standard and globally unique way within a message, could be incorporated into existing systems which do not currently have this capability. See section 3.1 [1984 X.400 Implementations], page 27, for an example of this.

However, some existing messaging systems do have mechanisms that parallel those of Body Part 15. Gateways capable of properly mapping/converting body part types are needed to provide for a high degree of interoperability between X.400 messaging systems and these existing systems. Figure 3.1 diagrams such a gateway in a generic fashion. Mapping of body parts is a critical responsibility of such a gateway. In order to convert corresponding body part types, the gateway also needs to "translate" the identifier that represents the body part type and decode/recode the application data. For this reason it is important that organizations defining new X.400 extended body part types, in addition to registering OIDs for Body Part 15, take the necessary steps to also identify the corresponding body part types in other messaging environments. See section 3.2 [Internet Mail – MIME], page 29, for an example of this.

Figure 3.1: Gateway's mapping of Body Parts

3.1 1984 X.400 Implementations

Applicability of Body Part 15 to 1984 X.400 messaging communities can be expressed in one of the following three scenarios:

1. 1984 messaging communities wishing to benefit from Body Part 15 feature without upgrading to 1988 systems.

2. Co-existing 1984 and 1988 messaging communities.

3. 1984 messaging communities migrating to 1988 X.400 systems.

Each of these are discussed in some detail in this section.

As mentioned earlier, the specific features required in a 1988 X.400 implementation to provide for Body Part 15 benefits are minimal. For this reason, in order to fully benefit from the Body Part 15 capability, it is recommended that X.400 PRMD Operators plan for transition to 1988 X.400.

3.1.1 1984 Messaging Communities

Through bilateral agreements between communicating parties, it is possible to use existing tags other than 15 for communication of extended body parts. For example tag 14 can be used for this purpose.

BilaterallyDefinedBodyPart Type (BP14)

The UnidentifiedBodyPart is included in the 1984 X.400 Implementor's Guide, Version 6, and is renamed as BilaterallyDefinedBodyPart in the 1988 X.400 series with the same tag and definition.

1988 X.420 defines bilaterally defined body part as:

> 7.3.10 Bilaterally Defined
>
> A **bilaterally defined** body part represents an information object whose semantics and abstract syntax are *bilaterally agreed* by the the IPM's originator and all of its potential recipients.

Typically it is the sender's responsibility to know whether it is in agreement with the receiver about what a bilaterally defined body part means. Anyone can use this body part for anything they want as long as they know that the receiver is capable of handling whatever it is that the part contains.

Any specific agreements on a particular usage of bilaterally defined body part by a community of users may potentially conflict with existing usage of bilaterally defined parts based on agreements by a different community of users.

Through bilateral agreements between communicating parties, it is possible to use Body Part 14 to exchange extended body parts. However, use of Body Part 14 for this purpose is discouraged by the 1988 and 1992 versions of X.400 recommendations and the use of Body Part 15 is recommended instead.

The OIW Implementor Agreements [?], make a number of recommendations for the transfer of binary data. Inclusion of extended body part information objects as Body Part 14 is not in conflict with these recommendations. There may be limitations to this use, however, as some existing 1984 user agents may have mechanisms for identification of information objects in Body Part 14 messages which may be incompatible with inclusion of extended body part types as Body Part 14.

3.1.2 1984/1988 Mixed Messaging Communities

The original 1984 X.400 specifications were developed without provisions for future extensions. Because of this limitation, a 1988 implementation cannot interwork with a 1984 implementation except by resembling the behavior of the 1984 implementation.[1] This process is called "downgrading".

Where 1984 and 1988 X.400 communities must coexist, PRMD Operators should attempt to procure 1988 X.400 implementations which handle "downgrading" of Body Part 15 to some type allowable under a 1984 system.

In cases where the Body Part 15 is the only element of 1988 X.400 functionality in the Content, 1988 implementations could "downgrade" the P22 content to P2 by converting encapsulated Body Part 15 objects to unidentified (BP 14) objects.

3.1.3 1984 Messaging Communities Migrating to 1988

Not all of the 1988 features need to be implemented for users to benefit from the Body Part 15 facility. Specifically only support for Body Part 15 in the protocol implementation and the UA capability registration features are needed. Other 1988 features such as Use of X.500, Distribution Lists, Message Store, Physical Delivery, Security, ... need not be implemented for the Body Part 15 benefits to be realized.

[1] See the companion "1984 to 1988 Migration" paper for details. Some of the relevant information from that paper is reproduced here.

In fact, including the Body Part 15 upgrade as an early stage of 1984 to 1988 migration planning is recommended. See the companion "88/84 Migration" paper for more information.

3.2 Internet Mail – MIME

The Internet is a collection of networks spanning the globe, with a large number of research, educational and commercial networks connected together into one global internet. The Internet serves as both a live testbed for on-going networking research and a daily communications tool for millions of users. It is a major tool in academic and industrial research in computer technology, physics, and astronomy, and increasingly in biological, social, and other sciences.[2]

Internet's research and engineering communities have been working on electronic mail technology since the early 1970's and have a great deal of practical experience in the area of electronic mail. Use of electronic mail services in Internet is very high and the level of e-mail connectivity to Internet has been growing rapidly. Since the introduction of X.400 (1984), there has been work ongoing for specifying mappings between X.400 MHS and Internet e-mail.

Recently, the Internet Engineering Task Force (IETF) has developed a document titled "Multipurpose Internet Mail Extensions" or MIME, RFC-1341,[3] [?]. MIME makes it possible to include, in a standardized way, arbitrary types of data objects within Internet e-mail messages. MIME imposes no special requirements on existing Internet message transfer systems and requires modifications only to User Agent software. Several MIME implementations are already in use. Modifications (patches) to most generally available e-mail software, for incorporation of the MIME features are also available.

MIME defines the structure for Internet message bodies through enhancements to the Content-Type field.[4] In X.400 specifications "content-type" is an identifier, on a message envelope, that identifies the content of the message. In MIME, the term "content-type" is used to identify an information object contained in the body of a message. In contrast to MIME's use of "content-type", X.400 uses the term "body part type". While reading the remainder of this section it is important to note this difference in terminology.

The `Content-Type` header field is used to specify the nature of the data in the

[2] "The Matrix", [?], contains additional information on this subject.

[3] RFCs (Request For Comments) are a series of Internet technical documents.

[4] Some of the information contained in this section has been taken from MIME and other related Internet RFCs.

body of an entity, by giving type and subtype identifiers, and by providing auxiliary information that may be required for certain types. After the type and subtype names, the remainder of the header field is simply a set of parameters, specified in an attribute/value notation.

In general, the top-level Content-Type is used to declare the general type of data, while the subtype specifies a specific format for that type of data. Subtype does not have a direct analogy in X.400. Thus, a Content-Type of ‘‘image/xyz” is enough to tell a user agent that the data is an image, even if the user agent has no knowledge of the specific image format ‘‘xyz”. Such information can be used, for example, to decide whether or not to show a user the raw data from an unrecognized subtype – such an action might be reasonable for unrecognized subtypes of text, but not for unrecognized subtypes of image or audio.

The ‘‘application” `Content-Type` is to be used for data which does not fit in any of the other categories, and particularly for data to be processed by mail-based uses of application programs. This is information which must be processed by an application before it is viewable or usable to a user. The current specification of MIME defines three subtypes: octet-stream, ODA, and Postscript.

3.2.1 MIME Registration of Content-Types

MIME defines a registration process which uses the Internet Assigned Numbers Authority (IANA) as a central registry for such values. See section G [MIME Registration of Content-type/subtype Values with IANA], page 71, for details about how IANA registration is accomplished. This information has been taken from RFC 1341. Parameters (i.e., numbers and keywords) used in protocols in the Internet community are assigned by the Internet Assigned Numbers Authority (IANA). Currently assigned values from several series of numbers used in Internet protocol implementations are documented in ‘‘Assigned Numbers”, RFC 1340, [**?**].

Here, we highlight some differences in the MIME’s content-type registration method and X.400’s Body Part 15 registration method.

MIME: All MIME Content-Types should be defined by IANA.

BP15: Not all Body Part 15 OIDs have to be assigned by the same registration authority.

MIME: Publication of MIME Content-types is accomplished through updates to the ‘‘Assigned Numbers” document.

BP15: As of this date, a Body Part 15 OID “publishing entity” does not exist. See section 4.2 [Body Part 15 OID Publication and Registration], page 34, for details of this issue.

3.2.2 Relationship between MIME and X.400

Since the introduction of X.400 (1984), there has been work ongoing for defining mappings between X.400 MHS and Internet e-mail. The early work on this topic focused primarily on the translation of envelope and headers.

Recently there has been work ongoing that focuses on the mapping of message bodies. Those interested in more information on this work should contact Internet Engineering Task Force MIME-MHS Interworking Working Group.

These documents have been specifically designed to provide optimal behavior for three different scenarios:

1. Allow a MIME user and an MHS user to exchange an arbitrary binary content.

2. Allow MIME content-types to “tunnel” through an MHS relay (that is, two MIME users can exchange content-types without loss through an MHS relay).

3. Allow MHS body parts to “tunnel” through a MIME relay (that is, two MHS users can exchange body parts without loss through a MIME relay).

To facilitate the mapping process, the Internet Assigned Numbers Authority (IANA) is expected to maintain a table termed the “IANA MHS/MIME Equivalence Table”. Once an entity has registered an OID to describe an MHS body part, it should ensure having a corresponding entry with the IANA. In practice, the corresponding content-type will be “application”, with an appropriate choice of sub-type and possible parameters.

Coordination in assignment of X.400 Body Part 15 OIDs and MIME content type/subtype is important. When implementors and vendors register OIDs and MIME content-types for their various objects they are strongly encouraged to specify their content formats such that conversion is as simple as possible. The simplest translation of byte copying is clearly desirable from the gateway implementation perspective.

Those interested in issues surrounding the MIME/X.400 message body mapping are encouraged to consult the Internet technical documents mentioned earlier in this section.

3.3 Proprietary Mail Systems

As mentioned earlier, there are two scenarios that best illustrate the relationship of Body Part 15 to existing proprietary mail systems.

1. Proprietary messaging systems that can transfer application data without regard to its type, but that are not capable of identifying the information type on a global basis.

2. Proprietary messaging systems that have mechanisms for identification of information types transferred through the messaging system.

In case of scenario (1), the Body Part 15 mechanism may be viewed as an enhancement which some implementors may be able to integrate into their existing proprietary messaging systems.

In case of scenario (2), gateways capable of body part mapping are required for interworking. Figure 3.1 diagrams such a gateway in a generic fashion.

In both scenarios, Application Program Interfaces (APIs) that accommodate processing of Body Part 15 OIDs play a significant role. For example, Microsoft has announced that MAPI will adopt the Body Part 15 structure and use of OIDs. This allows a Body Part 15 structured file to be transferred from an X.400 environment through a MAPI interface and retain full document characteristics.

Publication of Body Part 15 OIDs is needed so that they can be used in conjunction with e-mail APIs and enable the widespread integration of more applications into messaging systems.

In addition to e-mail systems, other systems (e.g., format translators, display systems) can benefit from a global mechanism for identification of file contents. Various operating systems have proprietary file type tagging capabilities. Other proposals for use of OIDs that make exchangeable files and documents self-identifying have also been made. The "Self-Identifying Files" proposal is based around placing a standard identifier in a standard location in all documents. That identifier can be the same as a Body Part 15 OID.

Chapter 4

Issues

This chapter focuses on issues and problems attendant to messaging transfer of application data such as the need for vendors to register application data types; potential problems of interoperability between different versions of messaging systems; the potential for difficulties when conversions take place, and the lack of a "Publishing House" to publish existing registered application data types.

In Chapter 5, (see section 5 [Recommendations], page 37), recommendations are made to address some of the issues and problems outlined in this section.

4.1 Body Part 15 OID Registration Process

The format of information objects to be communicated using the X.400 Body Part 15 facility is identified using the primitive data type, `OBJECT IDENTIFIER`, as defined in ISO 8824, [?]. See section B [Tutorial on Object Identifiers], page 51, for information on how `OBJECT IDENTIFIERS` are specified.

In order to guarantee that an Object Identifier is unique and unambiguous, it must be registered with a Registration Authority. In this section we focus on the specifics relating to Registration of OIDs for Body Part 15.

Currently, although a technical mechanism exists for identifying application data types in X.400 messages and there exist registration authorities who can issue unique identifiers, use of OIDs for communication of information objects using the Body Part 15 mechanism is not widespread.

A number of factors contribute to delays in widespread use of OIDs in Body Part 15. Given the existing status of the registration process, one of these contributing factors is that organizations who are best suited for registering the application data types (the owner of the application/document format) may not be familiar with the need for obtaining an OID for that format. It is hoped that the publication of this paper will raise the awareness of the need and benefits for application, document and messaging vendors to identify their formats with Body Part 15 OIDs. This will provide for better integration of applications and the messaging system (i.e. mail-enabled applications).

Another important contributing factor delaying the widespread use of OIDs in Body Part 15 is that presently there does not exist a widely adopted method to publicize the registration of application data types and corresponding identifiers. See section 4.2 [Body Part 15 OID Publication and Registration], page 34, for details of this issue.

The Body Part 15 mechanism enables exchange of typed information objects but does not necessarily impose a typing hierarchy on information objects. For example, a specific version of a PC based spread-sheet application data type may be defined as an extended body part type which identifies that information object to the messaging system but does not convey the hierarchy of "application-data/spread-sheet/PC-based/version-number" to the messaging system. It is possible to create a typing hierarchy using the Body Part 15 extension mechanism. For example, a well known entity (e.g., EMA, ANSI, NIST) can define an extended body part type (e.g., `application-data-body-part`) and then using the `Parameters` facility define sub-types and additional parameters. Presently, such a hierarchical typing mechanism has not been adopted. MIME defines a limited hierarchical typing mechanism which can be mimicked. See section 3.2 [Internet Mail – MIME], page 29, for more information.

To address the issues raised in this section, this paper proposes a number of recommendations. See section 5 [Recommendations], page 37, for details of these recommendations.

4.2 Body Part 15 OID Publication and Registration

Once an extended body part has been defined, its integration into messaging systems is possible. However, the specification of the newly defined body part must be made known to application developers, messaging implementors and users before it can be useful.

Although Body Part 15 object identifiers must be globally unique, not all Body Part 15 OIDs have to be assigned by the same registration authority. In fact, object identifiers for Body Part 15 will be assigned by many organizations, each of which has the choice of selecting its position in the object identifier tree.

The class of information object (for example Body Part 15), to which OIDs are assigned may play no role in the hierarchy of the components of the OID. Because of this, OIDs relating to Body Part 15 are spread out throughout the object identifier tree.

If all Body Part 15 OIDs were to be registered by a single registration authority, that registration authority would be able to prevent more than one OID from being assigned to the same exact format and also "publish" a list of all currently registered OIDs. Furthermore, many other aspects of the Body Part 15 OID registration process could be simplified in the above scenario. For example, if a well known entity were to function as the central registration authority for Body Part 15 OIDs, then the definition of Body Part 15 types can be accomplished through this central registration authority. Organizations wishing to define new Body Part 15 types can then go to this central registration authority to obtain OIDs and this central registration authority can publish these newly defined extended body part type definitions. All of this can be done in one place by the central registration authority.[1]

Presently, a well known central registration authority that can register all Body Part 15 OIDs does not exist.

For a number of reasons, it is highly desirable to have a list of all document/file types and their corresponding Body Part 15 OIDs. First, such a list can prevent more than one OID from being assigned to the same exact format. Also, by knowing the list of existing formats that can be communicated through the messaging system, software developers can accommodate processing the selection of formats that best suits them. It is also expected that by publishing such a list the use of extended body parts will become more widespread.

While Registration Authorities exist in the U.S. and other countries, presently no mechanism for publication of these Body Part 15 OIDs exists. To address this problem, this paper proposes a number of recommendations. See section 5 [Recommendations], page 37, for details of these recommendations.

4.3 Conversion

The 1988 X.400 specifications include provisions for message body part conversion. This conversion facility increases the level of transparent interoperability between mail senders and receivers.[2] For example, the sender can create a document in his/her favorite word processor A and mail it to the receiver who then receives it on a different

[1] See section 3.2.1 [MIME Registration of Content-Types], page 30, for an example of the central registration method.

[2] "EMA's 1988 X.400 Migration" white paper details many of the benefits and issues surrounding conversion. Some of these are reproduced here.

platform in his/her favorite word processor B.

Conversions may be performed based on Encoded Information Types (EIT) information in the envelope and UA capability registration.

The purpose of the Encoded Information Types in the message envelope is to allow the MTA to make decisions regarding the deliverability of a message or the need to invoke a conversion. By registering with the MTA the set of acceptable EITs, a Message Handling System (MHS) user ensures that he/she can process the messages that are delivered to him/her. Even more, the originator, upon receiving the receipt notification, can be assured that the recipient is, at least in theory, able to understand the message.

The specifics of how conversion of extended body parts is performed are beyond the scope of this paper. However, when acquiring messaging products PRMD Operators should pay special attention to the product's conversion capabilities and be sure that the specific conversions needed by the PRMD are supported. Also, since new convertible information objects (e.g., documents) and new versions of those information objects will continually emerge on the market, PRMD Operators should also inquire as to whether the product supports a generalized mechanism for configuring conversion. A generalized mechanism for configuring converter software modules "on the fly" so that arbitrary Body Part 15 to Body Part 15 object conversion can be performed provides a great deal of flexibility to PRMD Operators.

Chapter 5

Recommendations

While analyzing the issues surrounding the transfer of application data in this paper, the need for obtaining and publicizing OIDs which uniquely and unambiguously identify the format of information objects has become clear.

This section focuses on the steps that can be taken to facilitate the widespread use of externally defined application data types in messaging systems. This in turn will result in increased utility of the application data to its recipients.

5.1 Registering Extended Body Part Types

To address the issues and problems surrounding the extended body part OID registration (see section 4.1 [Body Part 15 OID Registration Process], page 33), the POC recommends that application, document and messaging vendors:

- **Identify the formats that should be defined as extended body part types.**

  The *owner* of an application format is best positioned to define an extended body part for that format so that its integration into messaging systems is possible. However, the ownership of an application format is not always obvious, and in these cases, the entity (e.g., company, organization, etc.) which specified the format should be considered the owner. In cases where the owner of the format is not interested in defining the extended body part for that format, other interested organizations should through consensus decide what secondary organization should have responsibility for defining the extended body part.

  Having more than one definition of the same exact format as multiple extended body parts is undesirable. Not only are multiple definitions unnecessary, but they may also complicate the development and management of both MTAs and UAs. Further, when companies issue procurement specifications for desired application types, multiple definitions may cause confusion, delays and errors. The absence of recognized extended body type publishing entities results in a higher probability of the same exact format being multiply defined. See section 5.2 [Publishing Extended Body Part OIDs], page 39. For these few reasons alone, it is extremely important to strive for global uniqueness in identifying file formats.

The primary criterion for identification of application formats which need to be defined as extended body part types should be the benefits of its integration into the messaging system.

In addition to vendors, some user organizations should consider defining extended body part types corresponding to their custom information formats (e.g., travel authorization, expense reports). These extended body part types can then be used in conjunction with UA implementations offering extensible configurations for automated processing of application data (see section 2.1 [About Application Integrated Messaging], page 9). Widespread publication of these extended body part types however, is probably unnecessary in most cases.

- **Identify the Registration Authority best suited for issuing the OIDs.**

  Not all Body Part 15 OIDs have to be assigned by the same registration authority. Each organization which wants to register an OID has the choice of selecting its position in the object identifier tree. In addition to reliability/recognition of the registration authority, the convenience and cost of registering your organization as an arc on the OID tree should be considered. In the US, ANSI is the top registration authority. Information about contacting ANSI is included in this paper. Choice of other registration authorities is possible.

- **Define the Extended Body Part and obtain the required OIDs.**

  The 1988 X.420 [**?**], mandates that Extended Body Parts shall:

  1. Use `OBJECT IDENTIFIER`s (OIDs) to uniquely identify the contents.
  2. Be defined by using the ASN.1 `EXTENDED-BODY-PART-TYPE` macro. See section A.3.1 [Externally defined body part], page 47.

  Therefore, definition of the extended body part must include the *complete* `EXTENDED-BODY-PART-TYPE` definition. The decision to use the `PARAMETERS` component should be given careful consideration. Omitting the `PARAMETERS` component makes the extended body part definition simpler.

Once the definition of the `EXTENDED-BODY-PART-TYPE` has been completed, OID assignments should be made.

See section 2.4.1 [Example Definition of an Extended Body Part Type], page 21, for a detailed example of how to accomplish both of the steps mentioned above. Here we present some of the more important points made in that example:

  - The organization defining the extended body part type should obtain a unique Numeric Name for their organization.
  - The organization should maintain an internal registration authority responsible for maintaining the organization's OID name space.

- **Publicize the complete definition of the extended body parts.**

  Communicate the complete specification of the newly defined extended body part to appropriate organizations. See section 5.2 [Publishing Extended Body Part OIDs], page 39.

- **Attempt to ensure coordinated definition of extended body part type and other similar systems.**

  See section 3.2.1 [MIME Registration of Content-Types], page 30, for an example of how X.400 body part definition and MIME Content-Type definitions can be coordinated.

- **Cooperate on integration of the newly defined extended body parts into messaging systems.**

  Once an extended body part has been defined its integration into messaging systems is possible. Such integration often requires cooperation amongst application developers, messaging implementors and users.

5.2 Publishing Extended Body Part OIDs

The specification of the newly defined body part type must be made known to application developers, messaging implementors and users before it can be useful. A Body Part 15 "publishing entity" can be used to publicize the specification of the newly defined body part types on a global basis. The need for publishing entities was identified in several places throughout this paper. At this time, such a publishing entity does not exist. Lack of such a publishing entity is viewed by many as an obstacle to realization of application integrated messaging.

Since not all Body Part 15 OIDs have to be assigned by the same registration authority, it is possible for the same exact format to be multiply defined. As described earlier, this is undesirable. By having a record of all the Body Part 15 OIDs in one place at the publishing entity, the probability of inadvertent multiple definitions will decrease.

The initial steps that can be taken to address this issue include:

- Identify Body Part 15 OIDs that have so far been registered.
- Identify the functions that such a publishing entity should perform.
- Identify the information to be collected and published.

These are discussed in some detail in this section.

Once an extended body part type publishing entity is in place, the "population" of defined extended body part types is expected to grow rapidly. A recent study by the PRMD Operators Committee has identified more than 150 application data types whose early integration into the messaging environment is expected.

It is recommended that any organization wishing to function as a Body Part 15 "publishing entity" do so consistent with the description outlined in this section. This section first identifies the responsibilities expected from a publication entity. Then, other related issues such as specifics of the information to be published are addressed.

5.2.1 Expected from a publishing entity

Responsibilities of extended body part type publisher include:

- Produce a "form" for collection of information to be maintained by the publisher. See section 5.2.2 [Extended body part type information to be collected], page 41, for details about the specifics of the information to be collected. The form that Electronic Mail Association has prototyped for gathering information about definition of extended body part type is included as an appendix to this document. See section F [Sample Form for Object Identifiers (OIDs)], page 69.
- Make the above mentioned "Extended body part type information collection form" available to organizations requesting publication of their newly defined body part type.
- Accept, date and file, the completed "Extended body part type information collection form".

- Be ready to make the complete list of known extended body part information available to those requesting it.

The following is *NOT* viewed as the responsibility of the publishing entity. The extended body part type publisher will *NOT*:

- Be responsible for correctness of information that is published beyond ensuring that the information provided on the "Extended body part type information collection form" is correctly published.
- Make extended body part type OID assignments.
- Publish application data information for which an extended body part type has not been defined.
- Ensure uniqueness of OIDs.

5.2.2 Extended body part type information to be collected

This section enumerates the specifics of information to be collected by the publishing entity for record keeping and publishing. A subset of this information can then be published.

- Extended Body Part Type Name
- Owner Information
  - Company Name.
  - Contact Name.
  - Contact Phone.
  - Contact E-Mail Address.
  - Date, extended body part type was defined.
  - ...
- Extended Body Part Type Information
  - Complete definition of the `EXTENDED-BODY-PART-TYPE` macro.

  - Complete `Data` OID.
  - Complete `Parameters` OID, if used.
  - Reference to Application Data Information
  - ...

- Application Data Information

  - Application data format specification. If the specification of the application data is an "open specification", an authorative reference to the specification should be included.
  - Application name or Document Type.
  - Version Number.
  - File Extension.
  - ...

- Security Considerations
- Other Miscellaneous Information

  - Date form received.
  - Date form modified.
  - Additional Comments.
  - ...

Appendix A

Tutorial on Messaging Transfer of Application Data

This section provides a basic overview of key aspects of the X.400 recommendations that relate to transfer of application data. In addition to this tutorial, other informational material is also included in this paper. See section B [Tutorial on Object Identifiers], page 51, for an overview of concepts relating to assignment of unique identifiers to information objects. See section D [Glossary of Terms], page 57, for explanation on terms used throughout this paper.

The tutorial material in this section specifically covers topics directly related to Body Part 15 and is quite condensed. Readers who are interested in more detailed explanations of messaging concepts should refer to other tutorials on X.400 messaging.

Some of the material included in this section is reproduced from the X.400 specifications.

A.1 Structure of messages

The basic structure of messages conveyed by the Message Transfer Service (MTS) is shown in Figure A.1. A message is made up of an envelope and a content. The envelope carries information that is used by the MTS when transferring the message. The content is the piece of information that the originating User Agent (UA) wishes delivered to one or more recipients.

Interpersonal messages (IP-messages) are a specialized class of messages specifically tailored for people oriented messaging.

A.2 Structure of Interpersonal Messages

The Message Transfer Service provides for exchange of interpersonal messages. The interpersonal message service (IPM service) is specifically designed to assist in people oriented communication. The specific content that is sent from one IPM UA to another is a result of an originator composing and sending a message, called an interpersonal message (IP-message).

The structure of an IP-message as it relates to the basic message structure of Message Handling Service (MHS) is shown in Figure A.2. The IP-message is conveyed with an envelope when being transferred through the MTS.

The IP-message contains information (e.g., to, cc, subject) provided by the user which is transformed by the IPM UA into the heading of the IP-message. The main information that the user wishes to communicate is contained within the body of the IP-message. In general, an IP-message body can consist of a number of body parts, each of which can be of a different encoded information type, such as voice, text, facsimile and graphics.

A.3 Types of body parts

IP-Messages sent and received in the IPM service can be composed of one or more body parts. Applicable body part types are defined in Recommendation X.420, [**?**], and comprise the following:

- IA5 text (tag [0]),
- Voice (tag [2]),
- G3 facsimile (tag [3]),
- G4 class 1 (tag [4]),
- Teletex (tag [5]),
- Videotex (tag [6]),
- Encrypted (tag [8]),
- Message (e.g., for a forwarded message) (tag [9]),
- Mixed mode (tag [11]),

- Bilaterally defined (tag [14]),
- Nationally defined (tag [7]),
- Externally defined (tag [15]).

In practice, IA5 text and Bilaterally define body parts are in common use while use of other body parts have not been as common.

A.3.1 Externally defined body part

An **externally defined** body part represents an information object whose semantics and abstract syntax are denoted by an object identifier which the body part carries. It has parameters and data components. The definition of `ExternallyDefinedBodyPart` is reproduced here.

```
ExternallyDefinedBodyPart ::= SEQUENCE {
    parameters        [0] ExternallyDefinedParameters OPTIONAL
    data              ExternallyDefinedData}

ExternallyDefinedParameters ::= EXTERNAL

ExternallyDefinedData ::= EXTERNAL
```

The `Parameters` and `Data` components are externals. (see clause 32 of Recommendation X.208, [?]). External type's distinguished values cannot be deduced from their characterization as external, but can be deduced from the encoding of such a value. The value does not even need to be described in ASN.1. The definition of `EXTERNAL` is reproduced here.

```
EXTERNAL ::= [UNIVERSAL 8] IMPLICIT SEQUENCE
{direct-reference       OBJECT IDENTIFIER OPTIONAL,
indirect-reference      INTEGER OPTIONAL,
data-value-descriptor   ObjectDescriptor OPTIONAL,
encoding                CHOICE
    {single-ASN1-type   [0] ANY,
    octet-aligned       [1] IMPLICIT OCTET STRING,
    arbitrary           [2] IMPLICIT BIT STRING}}
```

For both `parameters` and `data` of externally defined body part, direct-reference components of `EXTERNAL` shall be present, their indirect-reference and data-value-descriptor components absent.

On the basis of externally defined body part type, all body part types are divided into two important classes as follows:

basic Said of any body part type except externally defined.

extended Externally defined body part type as defined above.

Every extended body part type is defined by means of the `EXTENDED-BODY-PART-TYPE`. The definition of `EXTENDED-BODY-PART-TYPE` is reproduced here.

```
EXTENDED-BODY-PART-TYPE MACRO::=
BEGIN
   TYPE NOTATION  ::= Parameters Data
   VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)

   Parameters     ::=  "PARAMETERS" type "IDENTIFIED"
                       "BY" value(OBJECT IDENTIFIER) | empty
   Data           ::= "DATA" type
END
```

An instance of the macro's value notation defines the Object Identifier that appears as the `direct-reference` component of the `data` component of such an (Externally Defined) body part. The Object Identifier identifies the encoding rules for the body part.

If an Externally Defined body part has a `parameters` component, the Object Identifier in its direct-reference component is allocated at the same time and by the same naming authority as that in the `direct-reference` component of the `data` component.

A.4 Conversion between different encoded information types

The MTS provides conversion functions to allow IPM users to input messages in one encoded format, called encoded information type (EIT), and have them delivered in another EIT to cater to users with different terminal types. This capability is inherent in the IPM service, and increases the possibility of delivery by tailoring the message to the recipient's terminal capabilities. The EITs supported for the IPM service are defined in Recommendation X.420. IPM users have some control over the conversion process through various elements of service. These include the ability for a user to explicitly request the conversion required or as a default to let the MTS determine the need for, and type of, conversion performed. Users also have the ability to request that conversion not be performed or that conversion not be performed if loss of information will result. When the MTS performs conversion on a message, it informs the UA to whom the message is delivered that conversion took place and what the original EIT was. The conversion process for IP-messages can be performed on specific body parts if they are present in a message. The general aspects of conversion and the specific conversion rules for conversion between different EITs in the IPM service are detailed in Recommendation X.408, [?].

Appendix B

Tutorial on Object Identifiers

This section provides a basic overview of Object Identifiers as specified in ASN.1, [**?**].

The tutorial material in this section specifically covers aspects directly related to the topic of this paper and is quite condensed. Readers interested in more details should refer to ASN.1 specification or other tutorials on this subject.

Some of the material included in this section is reproduced from the ASN.1 specification.

B.1 Structure of Object Identifiers

The object identifier type in ASN.1 is referenced by the notation `OBJECT IDENTIFIER`.

The value of an object identifier comprises of an ordered list of `ObjIdComponent`s.

An `ObjIdComponent` may be one of three forms:

1. NameForm
2. NumberForm
3. NameAndNumberForm

The semantics of an object identifier value are defined by reference to an **object identifier tree**. An object identifier tree is a tree whose root corresponds to the ASN.1 standards and whose vertices correspond to administrative authorities responsible for allocating arcs from that vertex. Each arc of the tree is labelled by an object identifier component which is a numeric value. Each information object to be identified is allocated precisely one vertex (normally a leaf), and no other information object (of the same or a different type) is allocated to that same vertex. Thus an information object is uniquely and unambiguously identified by the sequence of numeric values (object identifier components) labelling the arcs in a path from the root to the vertex allocated to the information object.

In general, an information object is a class of information (for example, a file format), rather than an instance of such a class (for example, an individual file). It is thus the class of information (defined by some referenceble specification), rather than the piece of information itself, that is assigned a place in the tree.

B.2 Assignment of OBJECT IDENTIFIER component values

Three arcs are specified from the root node. The assignment of values and identifiers, and the authority for assignment of subsequent component values, are as follows:

Value	**Identifier**	**Authority for subsequent assignments**
0	ccitt	CCITT
1	iso	ISO
2	joint-iso-ccitt	See X.208 - Annex D

Four arcs are specified from the node identified by “iso”. The assignment of values and identifiers is

Value	**Identifier**	**Authority for subsequent assignments**
0	standard	See X.208 - Section B.4
1	registration-authority	See X.208 - Section B.5
2	member-body	See Below
3	identified-organization	See X.208 - Section B.7

The arcs immediately below “member-body” have values of three digit numeric country code, as specified in ISO 3166, that identifies the ISO Member Body in that country. The “NameForm” of object identifier component is not permitted with these identifiers.

In the US, ANSI is the authority for subsequent assignments.

Figure B.1 illustrates the OID tree described above. The dashed lines represent arcs that can be root hierarchies for object identifiers. It is possible for Body Part 15 OIDs to fall under any of the dashed lines. Organizations wishing to define a Body

Part 15 OID may be able to obtain an arc below any of the dashed lines. [1] Each circle represents a registration authority. The circles containing `X.208` also mention the relevant section. For example, (B.1) refers to X.208, Annex B, paragraph 1.

[1] Not necessarily immediately below.

Appendix C

Abbreviations

ADMD Administration management domain

ASN.1 Abstract Syntax Notation One

EIT Encoded information type

IPM Interpersonal messaging

IPMS Interpersonal messaging system

MD Management domain

MH Message handling

MHS Message handling system

MS Message store

MT Message transfer

MTA Message transfer agent

MTS Message transfer system

O/R Originator/recipient

OID Object Identifier

OSI Open system interconnection

PRMD Private management domain

UA User agent

Appendix D

Glossary of Terms

Some of the material included in this section is reproduced from ANNEX A of X.400 specification.

Note — The explanations given are not necessarily definitions in the strict sense.

Abstract Syntax Notation One (ASN.1)

A notation which both enables complicated types to be defined and also enables values of these types to be specified. This is done without determining the way an instance of this type is to be represented (by a sequence of octets) during transfer.

ASN.1 encoding rules

Rules which specify the representation during transfer of the value of any ASN.1 type; ASN.1 encoding rules enable information being transferred to be identified by the recipient as a specific value of a specific ASN.1 type.

arc

In the context of an object identifier tree, each arc is labelled by an object identifier component which is a numeric value.

basic service

In the context of message handling, the sum of features inherent in a service.

body

Component of a message. Other components are the heading and the envelope.

body part

Component of the body of a message.

content

In the context of message handling, an information object, part of a message, that the MTS neither examines nor modifies, except for conversion, during its conveyance of the message.

content type

In the context of message handling, an identifier, on a message envelope, that identifies the type (i.e. syntax and semantics) of the message content.

conversion

In the context of message handling, a transmittal event in which an MTA transforms parts of a message's content from one encoded information type to another, or alters a probe so it appears that the described messages were so modified.

delivery

In the context of message handling, a transmittal step in which an MTA conveys a message or report to the MS or UA of a potential recipient of the message or of the originator of the report's subject message or probe.

delivery report

In the context of message handling, a report that acknowledges delivery, non-delivery, export, or affirmation of the subject message or probe, or distribution list expansion.

encoded information type (EIT)

In the context of message handling, an identifier, on a message envelope, that identifies one type of encoded information represented in the message content. It identifies the medium and format (e.g., IA5 text, Group 3 facsimile) on an individual portion of the content.

envelope

In the context of message handling, an information object, part of a message, whose composition varies from one transmittal step to another and that variously identifies the message originator and potential recipients, documents its past and directs its subsequent conveyance by the MTS, and characterizes its content.

explicit conversion

In the context of message handling, a conversion in which the originator selects both the initial and final encoded information types.

extension of physical delivery address components

Standard attribute of a postal O/R address as a means to give further information about the point of physical delivery in a postal address, e.g., the name of a hamlet, or room and floor numbers in a large building.

extension of postal O/R address components

Standard attribute of a postal O/R address as a means to give further information to specify the addressee in a postal address, e.g. by organizational unit.

implicit conversion

In the context of message handling, a conversion in which the MTA selects both the initial and final encoded information types.

intercommunication

In the context of message handling, a relationship between services where one of the services is a message handling service, enabling the user of the message handling service to communicate with users of other services. Note - Examples are the intercommunication between the IPM service and the telex service, the intercommunication between message handling services and physical delivery services.

interpersonal messaging service

Messaging service between users belonging to the same management domain or to different management domains by means of message handling, based on the message transfer service.

IP-message

The content of a message in the IPM Service.

management domain (MD)

In the context of message handling, a set of messaging systems - at least one of which contains, or realizes, an MTA - at that is managed by a single organization. It is a primary building block used in the organizational construction of MHS. It refers to an organizational area for the provision of services. Note - A management domain may or may not necessarily be identical with a geographical area.

message

An instance of the primary class of information object conveyed by means of message transfer, and comprising an envelope and content.

message handling (MH)

A distributed information processing task that integrates the intrinsically related subtasks of message transfer and message storage.

message handling environment

The environment in which message handling takes place, comprising MHS, users, and distribution lists. The sum of all components of message handling systems. Note - Examples of components are:

```
-   message transfer agents,
-   user agents,
-   message stores,
-   access units,
-   users.
```

message handling service

Service provided by the means of message handling systems. Note 1 - Service may be provided through administration management domains or private management domains. Note 2 - Examples of message handling services are:

```
-   interpersonal messaging service (IPM service)
-   message transfer service (MT service).
```

message handling system (MHS)

The functional object, a component of the message handling environment, that conveys information objects from one party to another.

message store (MS)

The functional object, a component of MHS, that provides a single direct user with capabilities for message storage.

message transfer (MT)

The non-real-time – store-and-forward – carriage of information objects between parties using computers as intermediaries. It is one aspect of message handling.

message transfer agent (MTA)

A functional object, a component of the MTS, that actually conveys information objects to users and distribution lists.

message transfer service

Service that deals with the submission, transfer and delivery of messages for other messaging services.

message transfer system (MTS)

The functional object consisting of one or more message transfer agents which provides store-and-forward message transfer between user agents, message stores and access units.

messaging system

A computer system (possibly but not necessarily an open system) that contains, or realizes, one or more functional objects. It is a building block used in the physical construction of MHS.

naming authority

An authority responsible for the allocation of names.

non-delivery

In the context of message handling, a transmittal event in which an MTA determines that the MTS cannot deliver a message to one or more of its immediate recipients, or cannot deliver a report to the originator of its subject message or probe.

object identifier

A value (distinguished from all other such values) which is associated with an information object.

originator

In the context of message handling, the user (but not distribution list) that is the ultimate source of a message or probe.

private management domain (PRMD)

In the context of message handling, a management domain that comprises messaging system(s) managed by an organization other than an Administration.

report

In the context of message handling, an instance of a secondary class of information object conveyed by means of message transfer. It is generated by the MTS, it reports the outcome or progress of a message's or probe's transmittal to one or more potential recipients.

retrieval

In the context of message handling, a transmittal step in which a user's message store conveys a message or report to the user's UA. The user is an actual recipient of the message or the originator of the subject message or probe.

subject

In the context of message handling, that information, included as part of the header, that summarizes the content of the message as the originator has specified it.

user

In the context of message handling, a functional object (e.g., a person), a component of the message handling environment, that engages in (rather than provides) message handling and that is a potential source or destination for the information objects an MHS conveys.

vertex

In the context of an object identifier tree, each vertex corresponds to an administrative authority responsible for allocating arcs from that vertex.

X.208

CCITT Recommendations. Specification of abstract syntax notation one (ASN.1)

X.209

CCITT Recommendations. Specification of basic encoding rules for abstract syntax notation one (ASN.1)

X.400

CCITT Recommendations. Message handling systems and service overview.

X.402

CCITT Recommendations. Message handling systems: Overall architecture.

X.403

CCITT Recommendations. Message handling systems: Conformance testing.

X.407

CCITT Recommendations. Message handling systems: Abstract service definition conventions.

X.408

CCITT Recommendations. Message handling systems: Encoded information type conversion rules.

X.411

CCITT Recommendations. Message handling systems: Message transfer system: abstract service definition and procedures.

X.413

CCITT Recommendations. Message handling systems: Message store: Abstract-service definition.

X.419

CCITT Recommendations. Message handling systems: Protocol specifications.

X.420

CCITT Recommendations. Message handling systems: Interpersonal messaging system.

Appendix E

ANSI Registration Authority Contact

Registration Authority Contact


```
American National Standards Institute
1430 Broadway
New York, New York 10018
USA

Telephone     +1 212-642-4976

Fax           +1 212-398-0023

Telex         42 42 96 ANSI UI
```


ANSI 1992 Fee Schedule

```
                                         One-time fee (in U.S. $)
                                         -----------------------
- Registration fee for both forms                2,500
  (numeric and alphanumeric)
```

```
- Registration fee for numeric name                    1,000

- Registration fee for alphanumeric name               1,500
  (numeric name previously assigned)

- Register update fee                                    100

- Inquiry fee (per item)                                 100
```

Appendix F

Sample Form for Object Identifiers (OIDs)

Appendix G

MIME Registration of Content-type/subtype Values with IANA

The following section has been reproduced from RFC 1341, [?].

```
Note that MIME is generally expected to be extended by subtypes.  If a
new fundamental top-level type is needed, its specification should be
published as an RFC or submitted in a form suitable to become an RFC,
and be subject to the Internet standards process.

     To:  IANA@isi.edu
     Subject:  Registration of new MIME content-type/subtype

     MIME type name:

     (If the above is not an existing top-level MIME type,
     please explain why an existing type cannot be used.)

     MIME subtype name:
```

```
Required parameters:

Optional parameters:

Encoding considerations:

Security considerations:

Published specification:

(The published specification must be an Internet RFC or
RFC-to-be if a new top-level type is being defined, and
must be a publicly available specification in any
case.)

Person & email address to contact for further
information:
```

Concept Index

1

A

B

C

D

E

G

I

M

N

O

P

R

S