

Network Working Group  
Request for Comments: 2425  
Category: Standards Track

T. Howes  
M. Smith  
Netscape Communications Corp.  
F. Dawson  
Lotus Development Corporation  
September 1998

## A MIME Content-Type for Directory Information

### Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

### 1. Abstract

This document defines a MIME Content-Type for holding directory information. The definition is independent of any particular directory service or protocol. The text/directory Content-Type is defined for holding a variety of directory information, for example, name, or email address, or logo. The text/directory Content-Type can also be used as the root body part in a multipart/related Content-Type for handling more complicated situations, especially those in which non-textual information that already has a natural MIME representation, for example, a photograph or sound, is to be represented.

The text/directory Content-Type defines a general framework and format for holding directory information in a simple "type:value" form. We refer to "type" in this context meaning a property or attribute with which the value is associated. Mechanisms are defined to specify alternate languages, encodings and other meta-information. This document also defines the procedure by which particular formats, called profiles, for carrying application-specific information within a text/directory Content-Type can be defined and registered, and the conventions such formats must follow. It is expected that other documents will be produced that define such formats for various applications (e.g., white pages).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119].

## 2. Table of Contents

Status of the Memo.....	1
Copyright Notice.....	1
1. Abstract.....	1
2. Table of Contents.....	2
3. Need for a MIME Directory Type.....	3
4. Overview.....	4
5. The text/directory Content-Type.....	4
5.1. MIME media type name.....	4
5.2. MIME subtype name.....	5
5.3. Required parameters.....	5
5.4. Optional parameters.....	5
5.5. Encoding considerations.....	5
5.6. Security considerations.....	6
5.7. Interoperability considerations.....	6
5.8. Published specification.....	6
5.8.1. Line delimiting and folding.....	6
5.8.2. ABNF content-type definition.....	7
5.8.3. Pre-defined Parameters.....	9
5.8.4. Pre-defined Value Types.....	11
5.9. Applications which use this media type.....	14
5.10. Additional information.....	14
5.11. Person & email address to contact for further information..	14
5.12. Intended usage.....	14
5.13. Author/Change controller.....	15
6. Predefined Types.....	15
6.1. SOURCE Type Definition.....	15
6.2. NAME Type Definition.....	16
6.3. PROFILE Type Definition.....	16
6.4. BEGIN Type Definition.....	17
6.5. END Type Definition.....	17
7. Use of the multipart/related Content-Type.....	18
8. Examples.....	18
8.1. Example 1.....	19
8.2. Example 2.....	19
8.3. Example 3.....	20
8.4. Example 4.....	21
9. Registration of new profiles.....	22
9.1. Define the profile.....	22
9.2. Post the profile definition.....	23
9.3. Allow a comment period.....	23
9.4. Submit the profile for approval.....	23
10. Profile Change Control.....	23

11.	Registration of new types.....	24
11.1.	Define the type.....	24
11.2.	Post the type definition.....	25
11.3.	Allow a comment period.....	25
11.4.	Submit the type for approval.....	25
12.	Type Change Control.....	25
13.	Registration of new parameters.....	26
13.1.	Define the parameter.....	26
13.2.	Post the parameter definition.....	27
13.3.	Allow a comment period.....	27
13.4.	Submit the parameter for approval.....	27
14.	Parameter Change Control.....	28
15.	Registration of new value types.....	28
15.1.	Define the value type.....	28
15.2.	Post the value type definition.....	29
15.3.	Allow a comment period.....	29
15.4.	Submit the value type for approval.....	29
16.	Security Considerations.....	30
17.	Acknowledgements.....	30
18.	References.....	30
19.	Authors' Addresses.....	32
20.	Full Copyright Statement.....	33

### 3. Need for a MIME Directory Type

For purposes of this document, a directory is a special-purpose database that contains typed information. A directory usually supports both read and search of the information it contains, and can support creation and modification of the information as well. Directory information is usually accessed far more often than it is updated. Directories can be local or global in scope. They can be distributed or centralized. The information they contain can be replicated, with weak or strong consistency requirements.

There are several situations in which users of Internet mail might wish to exchange directory information: the email analogy of a "business card" exchange; the conveyance of directory information to a user having only email access to the Internet; the provision of machine-parseable address information when purchasing goods or services over the Internet; etc. As MIME [RFC-2045, RFC-2046] is used increasingly by other protocols, most notably HTTP, it can also be useful for these protocols to carry directory information in MIME format. Such a format, for example, could be used to represent URC (uniform resource characteristics) information about resources on the World Wide Web, or to provide a rudimentary directory service over HTTP.

#### 4. Overview

The scheme defined here for representing directory information in a MIME Content-Type has two parts. First, the text/directory Content-Type is defined for use in holding directory information within a single body part, for example name, title, or email address. In its simplest form, the format uses a "type:value" approach, which should be easily parseable by existing MIME implementations and understandable by users. More complicated situations can be represented also. This document defines the general form the information in the Content-Type should have, and the procedure by which specific types and values (properties) for particular applications can be defined. The framework is general enough to handle information from any number of end directory services, including LDAP [RFC-1777, RFC-1778], WHOIS++ [RFC-1835], and X.500 [X500].

Directory entries can include far more than just textual information. Some such information (e.g., an image or sound) overlaps with predefined MIME Content-Types. In these cases it can be desirable to include the information in its well-known MIME format. This situation is handled by using a multipart/related Content-Type as defined in [RFC-2112]. The root component of this type is a text/directory body part specifying any in-line information, and for information contained in other Content-Types, the Content-IDs (in URI form) of those parts.

In some applications, it can be useful to include a pointer (e.g, a URI) to some directory information rather than the information itself. This document defines a general mechanism for accomplishing this.

#### 5. The text/directory Content-Type

The text/directory Content-Type is used to hold basic directory information and URIs referencing other information, including other MIME body parts holding supplementary or non-textual directory information, such as an image or sound. It is defined as follows, using the MIME media type registration template from [RFC-2048].

To: ietf-types@uninett.no  
Subject: Registration of MIME media type text/directory

##### 5.1. MIME media type name

MIME media type name: text

## 5.2. MIME subtype name

MIME subtype name: directory

## 5.3. Required parameters

Required parameters: charset

The "charset" parameter is as defined in [\[RFC-2046\]](#) for other body parts. It is used to identify the default character set used within the body part.

## 5.4. Optional parameters

Optional parameters: profile

The "profile" parameter is used to convey the type(s) of entity(ies) to which the directory information pertains and the likely set of information associated with the entity(ies). It is intended only as a guide to applications interpreting the information contained within the body part. It SHOULD NOT be used to exclude or require particular pieces of information unless a profile definition specifically calls for this behavior. Unless specifically forbidden by a particular profile definition, a text/directory content type can contain arbitrary attribute/value pairs.

The value of the "profile" parameter is defined as follows. Profile names are case insensitive (i.e., the profile name "vCard" is the same as "VCARD" and "vcCard" and "vcArD").

profile = x-name / iana-token

x-name = "x-" 1\*(ALPHA / DIGIT / "-")  
; Names beginning with "x-" or "X-" are  
; reserved for experimental use not intended for released  
; products, or for use in bilateral agreements.

iana-token = <a publicly-defined extension token, registered  
with IANA, as specified in [Section 9](#) of this  
document>

## 5.5. Encoding considerations

The default encoding is 8bit. Otherwise, as specified by the Content-Transfer-Encoding header field.

### 5.6. Security considerations

Directory information can be public or it can be protected from unauthorized access by the directory service in which it resides. Once the information leaves its native service, there can be no guarantee that the same care will be taken by all services handling the information. Furthermore, this specification defines no access control mechanism by which information can be protected, or by which access control information can be conveyed. Note that the integrity and privacy of a text/directory body part can be protected by enclosing it within an appropriate MIME-based security mechanism.

### 5.7. Interoperability considerations

In order to make sense of directory information, applications must share a common understanding of the types of information contained within the Content-Type (the directory schema). This schema information is not defined in this document, but rather in companion documents (e.g., [MIME-VCARD]) that follow the requirements specified in this document, or in bilateral agreements between communicating parties.

### 5.8. Published specification

The text/directory Content-Type contains directory information, typically pertaining to a single directory entity or group of entities. The content consists of one or more lines in the format given below.

#### 5.8.1. Line delimiting and folding

Individual lines within the MIME text/directory Content Type body are delimited by the [RFC-822] line break, which is a CRLF sequence (ASCII decimal 13, followed by ASCII decimal 10). Long logical lines of text can be split into a multiple-physical-line representation using the following folding technique.

A logical line MAY be continued on the next physical line anywhere between two characters by inserting a CRLF immediately followed by a single white space character (space, ASCII decimal 32, or horizontal tab, ASCII decimal 9). At least one character must be present on the folded line. Any sequence of CRLF followed immediately by a single white space character is ignored (removed) when processing the content type. For example the line:

DESCRIPTION:This is a long description that exists on a long line.

Can be represented as:

DESCRIPTION:This is a long description  
that exists on a long line.

It could also be represented as:

DESCRIPTION:This is a long descrip  
tion that exists o  
n a long line.

The process of moving from this folded multiple-line representation of a type definition to its single line representation is called unfolding. Unfolding is accomplished by regarding CRLF immediately followed by a white space character (namely HTAB ASCII decimal 9 or SPACE ASCII decimal 32) as equivalent to no characters at all (i.e., the CRLF and single white space character are removed).

#### 5.8.2. ABNF content-type definition

The following ABNF uses the notation of [RFC 2234](#), which also defines CRLF, WSP, DQUOTE, VCHAR, ALPHA, and DIGIT. After the unfolding of any folded lines as described above, the syntax for a line of this content type is as follows:

```
contentline = [group "."] name *("; " param) ":" value CRLF
; When parsing a content line, folded lines MUST first
; be unfolded according to the unfolding procedure
; described above.
; When generating a content line, lines longer than 75
; characters SHOULD be folded according to the folding
; procedure described above.

group       = 1*(ALPHA / DIGIT / "-")

name        = x-name / iana-token

iana-token  = 1*(ALPHA / DIGIT / "-")
; identifier registered with IANA

x-name      = "x-" 1*(ALPHA / DIGIT / "-")
; Names that begin with "x-" or "X-" are
; reserved for experimental use, not intended for released
; products, or for use in bilateral agreements.

param       = param-name "=" param-value *(", " param-value)

param-name  = x-name / iana-token

param-value = ptext / quoted-string
```

```
ptext  = *SAFE-CHAR

value = *VALUE-CHAR
      / valuespec      ; valuespec defined in section 5.8.4

quoted-string = DQUOTE *QSAFE-CHAR DQUOTE

NON-ASCII     = %x80-FF
      ; use restricted by charset parameter
      ; on outer MIME object (UTF-8 preferred)

QSAFE-CHAR    = WSP / %x21 / %x23-7E / NON-ASCII
      ; Any character except CTLs, DQUOTE

SAFE-CHAR     = WSP / %x21 / %x23-2B / %x2D-39 / %x3C-7E / NON-ASCII
      ; Any character except CTLs, DQUOTE, ";", ":", ",", " "

VALUE-CHAR    = WSP / VCHAR / NON-ASCII
      ; any textual character
```

A line that begins with a white space character is a continuation of the previous line, as described above. The white space character and immediately preceeding CRLF should be discarded when reconstructing the original line. Note that this line-folding convention differs from that found in [RFC 822](#), in that the sequence <CRLF><WSP> found anywhere in the content indicates a continued line and should be removed.

Various type names and the format of the corresponding values are defined as specified in [Section 11](#). Specifications MAY impose ordering on the type constructs within a body part, though none is required by default. The various x-name constructs are used for bilaterally-agreed upon type names, parameter names and parameter values, or for use in experimental settings.

Type names and parameter names are case insensitive (e.g., the type name "fn" is the same as "FN" and "Fn"). Parameter values MAY be case sensitive or case insensitive, depending on their definition.

The group construct is used to group related attributes together. The group name is a syntactic convention used to indicate that all type names prefaced with the same group name SHOULD be grouped together when displayed by an application. It has no other significance. Implementations that do not understand or support grouping MAY simply strip off any text before a "." to the left of the type name and present the types and values as normal.



Each attribute defined in the text/directory body MAY have multiple values, if allowed in the definition of the profile in which the attribute is used. The general rule for encoding multi-valued items is to simply create a new content line for each value (including the type name). However, it should be noted that some value types support encoding multiple values in a single content line by separating the values with a comma ",". This approach has been taken for several of the content types defined below (date, time, integer, float), for space-saving reasons.

### 5.8.3. Pre-defined Parameters

The following parameters and value types are defined for general use.

```
predefined-param = encodingparm
                  / valuetypeparm
                  / languageparm
                  / contextparm

encodingparm = "encoding" "=" encodingtype

encodingtype = "b"           ; from RFC 2047
               / iana-token  ; registered as described in
                           ; section 15 of this document

valuetypeparm = "value" "=" valuetype

valuetype = "uri"           ; genericurl from section 5 of RFC 1738
           / "text"
           / "date"
           / "time"
           / "date-time" ; date time
           / "integer"
           / "boolean"
           / "float"
           / x-name
           / iana-token  ; registered as described in
                           ; section 15 of this document

languageparm = "language" "=" Language-Tag
               ; Language-Tag is defined in section 2 of RFC 1766

contextparm = "context" "=" context

context = x-name
         / iana-token
```

The "language" type parameter is used to identify data in multiple languages. There is no concept of "default" language, except as specified by any "Content-Language" MIME header parameter that is present. The value of the "language" type parameter is a language tag as defined in [Section 2 of \[RFC-1766\]](#).

The "context" type parameter is used to identify a context (e.g., a protocol) used in interpreting the value. This is used, for example, in the "source" type, defined below.

The "encoding" type parameter is used to specify an alternate encoding for a value. If the value contains a CRLF, it must be encoded, since CRLF is used to separate lines in the content-type itself. Currently, only the "b" encoding is supported.

The "b" encoding can also be useful for binary values that are mixed with other text information in the body part (e.g., a certificate). Using a per-value "b" encoding in this case leaves the other information in a more readable form. The encoded base 64 value can be split across multiple physical lines in the content type by using the line folding technique described above.

The Content-Transfer-Encoding header field is used to specify the encoding used for the body part as a whole. The "encoding" type parameter is used to specify an encoding for a particular value (e.g., a certificate). In this case, the Content-Transfer-Encoding header might specify "8bit", while the one certificate value might specify an encoding of "b" via an "encoding=b" type parameter.

The Content-Transfer-Encoding and the encodings of individual types given by the "encoding" type parameter are independent of one another. When encoding a text/directory body part for transmission, individual type encodings are performed first, then the entire body part is encoded according to the Content-Transfer-Encoding. When decoding a text/directory body part, the Content-Transfer-Encoding is decoded first, and then any individual types with an "encoding" type parameter are decoded.

The "value" parameter is optional, and is used to identify the value type (data type) and format of the value. The use of these predefined formats is encouraged even if the value parameter is not explicitly used. By defining a standard set of value types and their formats, existing parsing and processing code can be leveraged.

Including the value type explicitly as part of each property provides an extra hint to keep parsing simple and support more generalized applications. For example a search engine would not have to know the particular value types for all of the items for which it is

searching. Because the value type is explicit in the definition, the search engine could look for dates in any item type and provide results that can still be interpreted.

#### 5.8.4. Pre-defined Value Types

The format for values corresponding to the predefined valuetype specifications given above are defined.

```
valuespec = text-list
           / genericurl      ; from section 5 of RFC 1738
           / date-list
           / time-list
           / date-time-list
           / boolean
           / integer-list
           / float-list
           / iana-valuespec

text-list = *TEXT-LIST-CHAR *(", " *TEXT-LIST-CHAR)

TEXT-LIST-CHAR = "\\\" / "\", " / "\n"
                / <any VALUE-CHAR except , or \ or newline>
                ; Backslashes, newlines, and commas must be encoded.
                ; \n or \N can be used to encode a newline.

date-list = date *(", " date)

time-list = time *(", " time)

date-time-list = date "T" time *(", " date "T" time)

boolean = "TRUE" / "FALSE"

integer-list = integer *(", " integer)

integer = [sign] 1*DIGIT

float-list = float *(", " float)

float = [sign] 1*DIGIT [ "." 1*DIGIT ]

sign = "+" / "-"

date = date-fullyear ["-"] date-month ["-"] date-mday

date-fullyear = 4 DIGIT
```

```
date-month = 2 DIGIT      ;01-12

date-mday = 2 DIGIT      ;01-28, 01-29, 01-30, 01-31
                  ;based on month/year

time = time-hour [":" ] time-minute [":" ] time-second [time-secfrac]
      [time-zone]

time-hour = 2 DIGIT      ;00-23

time-minute = 2 DIGIT    ;00-59

time-second = 2 DIGIT    ;00-60 (leap second)

time-secfrac = "," 1*DIGIT

time-zone = "Z" / time-numzone

time-numzone = sign time-hour [":" ] time-minute

iana-valuespec = <a publicly-defined valuetype format, registered
                  with IANA, as defined in section 15 of this
                  document>
```

Some specific notes on the value types and formats:

"text": The "text" value type should be used to identify values that contain human-readable text. The character set and language in which the text is represented is controlled by the charset content-header and the language type parameter and content-header.

Examples for "text":

```
    this is a text value
    this is one value,this is another
    this is a single value\, with a comma encoded
```

A formatted text line break in a text value type MUST be represented as the character sequence backslash (ASCII decimal 92) followed by a Latin small letter n (ASCII decimal 110) or a Latin capital letter N (ASCII decimal 78), that is "\n" or "\N".

For example a multiple line DESCRIPTION value of:

```
Mythical Manager
Hyjinx Software Division
BabsCo, Inc.
```

could be represented as:

DESCRIPTION:Mythical Manager\nHyjinx Software Division\nBabsCo\, Inc.\n

demonstrating the \n literal formatted line break technique, the CRLF-followed-by-space line folding technique, and the backslash escape technique.

"uri": The "uri" value type should be used to identify values that are referenced by a URI (including a Content-ID URI), instead of encoded in-line. These value references might be used if the value is too large, or otherwise undesirable to include directly. The format for the URI is as defined in [RFC 1738](#).

Examples for "uri":

<http://www.foobar.com/my/picture.jpg>  
ldap://ldap.foobar.com/cn=babs%20jensen

"date", "time", and "date-time": Each of these value types is based on a subset of the definitions in ISO 8601 standard. Profiles MAY place further restrictions on "date" and "time" values. Multiple "date" and "time" values can be specified using the comma-separated notation, unless restricted by a profile.

Examples for "date":

1985-04-12  
1996-08-05,1996-11-11  
19850412

Examples for "time":

10:22:00  
102200  
10:22:00.33  
10:22:00.33Z  
10:22:33,11:22:00  
10:22:00-08:00

Examples for "date-time":

1996-10-22T14:00:00Z  
1996-08-11T12:34:56Z  
19960811T123456Z  
1996-10-22T14:00:00Z,1996-08-11T12:34:56Z

"boolean": The "boolean" value type is used to express boolean values. These values are case insensitive.

Examples: TRUE

false  
True

"integer": The "integer" value type is used to express signed integers in decimal format. If sign is not specified, the value is assumed positive "+". Multiple "integer" values can be specified using the comma-separated notation, unless restricted by a profile.

Examples: 1234567890  
          -1234556790  
          +1234556790,432109876

"float": The "float" value type is used to express real numbers. If sign is not specified, the value is assumed positive "+". Multiple "float" values can be specified using the comma-separated notation, unless restricted by a profile.

Examples: 20.30  
          1000000.0000001  
          1.333,3.14

#### 5.9. Applications which use this media type

Applications which use this media type: Various

#### 5.10. Additional information

Additional information: None

#### 5.11. Person & email address to contact for further information

Tim Howes  
Netscape Communications Corp.  
501 East Middlefield Rd.  
Mountain View, CA 94041  
USA  
howes@netscape.com  
+1 415 937 3419

#### 5.12. Intended usage

Intended usage: COMMON

### 5.13. Author/Change controller

Tim Howes  
Netscape Communications Corp.  
501 East Middlefield Rd.  
Mountain View, CA 94041  
USA  
howes@netscape.com  
+1 415 937 3419

Mark Smith  
Netscape Communications Corp.  
501 East Middlefield Rd.  
Mountain View, CA 94041  
USA  
mcs@netscape.com  
+1 415 937 3477

Frank Dawson  
Lotus Development Corporation  
6544 Battleford Drive  
Raleigh, NC 27613-3502  
USA  
frank\_dawson@lotus.com  
+1-919-676-9515

## 6. Predefined Types

The following types are generally useful regardless of the profile being carried and are defined below using the text/directory MIME type registration template defined in [Section 11.1](#) of this document. These types MAY be included in any profile, unless explicitly forbidden in the profile definition.

### 6.1. SOURCE Type Definition

To: ietf-mime-direct@imc.org  
Subject: Registration of text/directory MIME type SOURCE

Type name: SOURCE

Type purpose: To identify the source of directory information contained in the content type.

Type encoding: 8bit

Type valuetype: uri

Type special notes: The SOURCE type is used to provide the means by which applications knowledgeable in the given directory service protocol can obtain additional or more up-to-date information from the directory service. It contains a URI as defined in [RFC-1738] and/or other information referencing the directory entity or entities to which the information pertains. When directory information is available from more than one source, the sending entity can pick what it considers to be the best source, or multiple SOURCE types can be included. The interpretation of the value for a SOURCE type can depend on the setting of the CONTEXT type parameter. The value of the CONTEXT type parameter MUST be compatible with the value of the uri prefix.

Type example:

```
SOURCE;CONTEXT=LDAP:ldap://ldap.host/cn=Babs%20Jensen,  
%20o=Babsco,%20c=US
```

## 6.2. NAME Type Definition

To: ietf-mime-direct@imc.org  
Subject: Registration of text/directory MIME type NAME

Type name: NAME

Type purpose: To identify the displayable name of the directory entity to which information in the content type pertains.

Type encoding: 8bit

Type valuetype: text

Type special notes: The NAME type is used to convey the display name of the entity to which the directory information pertains.

Type example:

```
NAME:Babs Jensen's Contact Information
```

## 6.3. PROFILE Type Definition

To: ietf-mime-direct@imc.org  
Subject: Registration of text/directory MIME type PROFILE

Type name: PROFILE

Type purpose: To identify the type of directory entity to which information in the content type pertains.

Type encoding: 8bit



Type valuetype: A profile name, registered as described in [Section 9](#) of this document or bilaterally agreed upon as described in [Section 5](#).

Type special notes: The PROFILE type is used to convey the type of the entity to which the directory information in the rest of the body part pertains. It should be the same as the "profile" header parameter, if present.

Type example:

PROFILE:vCard

#### 6.4. BEGIN Type Definition

To: ietf-mime-direct@imc.org

Subject: Registration of text/directory MIME type BEGIN

Type name: BEGIN

Type purpose: To denote the beginning of a syntactic entity within a text/directory content-type.

Type encoding: 8bit

Type valuetype: text, containing a profile name, registered as described in [Section 9](#) of this document or bilaterally-agreed upon as described in [Section 5](#).

Type special notes: The BEGIN type is used in conjunction with the END type to delimit a profile containing a related set of properties within an text/directory content-type. This construct can be used instead of or in addition to wrapping separate sets of information inside additional MIME headers. It is provided for applications that wish to define content that can contain multiple entities within the same text/directory content-type or to define content that can be identifiable outside of a MIME environment.

Type example:

BEGIN:VCARD

#### 6.5. END Type Definition

To: ietf-mime-direct@imc.org

Subject: Registration of text/directory MIME type END

Type name: END

Type purpose: To denote the end of a syntactic entity within a text/directory content-type.

Type encoding: 8bit

Type valuetype: text, containing a profile name, registered as described in [Section 9](#) of this document or bilaterally-agreed upon as described in [Section 5](#).

Type special notes: The END type is used in conjunction with the BEGIN type to delimit a profile containing a related set of properties within an text/directory content-type. This construct can be used instead of or in addition to wrapping separate sets of information inside additional MIME headers. It is provided for applications that wish to define content that can contain multiple entities within the same text/directory content-type or to define content that can be identifiable outside of a MIME environment.

Type example:

END: VCARD

## 7. Use of the multipart/related Content-Type

The multipart/related Content-Type can be used to hold directory information comprised of both text and non-text information or directory information that already has a natural MIME representation. The root body part within the multipart/related body part is specified as defined in [\[RFC-2112\]](#) by a "start" parameter, or it is the first body part in the absence of such a parameter. The root body part must have a Content-Type of "text/directory". This part holds inline information and makes reference to subsequent body parts holding additional text or non-text directory information via their Content-ID URIs as explained in [Section 5](#).

The body parts referred to do not have to be in any particular order, except as noted above for the root body part.

## 8. Examples

The following examples are for illustrative purposes only and are not part of the definition.

### 8.1. Example 1

The first example illustrates simple use of the text/directory Content-Type. Note that no "profile" parameter is given, so an application may not know what kind of directory entity the information applies to. Note also the use of both hypothetical official and bilaterally agreed upon types.

```
From: Whomever@wherever.com
To: Someone@somewhere.com
Subject: whatever
MIME-Version: 1.0
Message-ID: <id1@host.net>
Content-Type: text/directory
Content-ID: <id2@host.com>
```

```
cn:Babs Jensen
cn:Barbara J Jensen
sn:Jensen
email:babs@umich.edu
phone:+1 313 747-4454
x-id:1234567890
```

### 8.2. Example 2

The next example illustrates the use of the Quoted-Printable transfer encoding defined in [RFC 2045] to include non-ASCII character in some of the information returned, and the use of the optional "name" and "source" types. It also illustrates the use of an "encoding" type parameter to encode a certificate value in "b". A "vCard" profile [MIME-VCARD] is used for the example.

```
Content-Type: text/directory;
    charset="iso-8859-1";
    profile="vCard"
Content-ID: <id3@host.com>
Content-Transfer-Encoding: Quoted-Printable
```

```
begin:VCARD
source:ldap://cn=bjorn%20Jensen, o=university%20of%20Michigan, c=US
name:Bjorn Jensen
fn:Bj=F8rn Jensen
n:Jensen;Bj=F8rn
email;type=internet:bjorn@umich.edu
tel;type=work,voice,msg:+1 313 747-4454
key;type=x509;encoding=B:dGhpcyBjb3VsZCBiZSAKbXkgY2VydGlmaWNhdGUK
end:VCARD
```

## 8.3. Example 3

The next example illustrates the use of multi-valued type parameters, the "language" type parameter, the "value" type parameter, folding of long lines, the \n encoding for formatted lines, attribute grouping, and the inline "b" encoding. A "vCard" profile [[MIME-VCARD](#)] is used for the example.

```
Content-Type: text/directory; profile="vcard"; charset=iso-8859-1
Content-ID: <id3@host.com>
Content-Transfer-Encoding: Quoted-Printable
```

```
begin:vcard
source:ldap://cn=Meister%20Berger,o=Universitaet%20Goerlitz,c=DE
name:Meister Berger
fn:Meister Berger
n:Berger;Meister
bday;value=date:1963-09-21
o:Universit=E6t G=F6rlitz
title:Mayor
title;language=de;value=text:Burgermeister
note:The Mayor of the great city of
    Goerlitz in the great country of Germany.
email;internet:mb@goerlitz.de
home.tel;type=fax,voice,msg:+49 3581 123456
home.label:Hufenshlagel 1234\n
    02828 Goerlitz\n
    Deutschland
key;type=X509;encoding=b:MIICajCCAdOgAwIBAgICBEUwDQYJKoZIhvcNAQEEBQ
AwdzELMAkGA1UEBhMCVVMxLDAqBgNVBAoTIO5ldHNjYXB1IENvbW11bmljYXRpb25zI
ENvcnBvcmlF0aW9uMRwwGgYDVQQLEXNjb2Vzcm1hdGlvbiBTeXN0ZW1zMRRwGgYDVQQD
ExNyY290Y2EubmV0c2NhcnGUuY29tMB4XDtk3MDYwNjE5NDc1OVoxDTk3MTIwMzE5NDc
1OVowgYkxCzAJBgNVBAYTA1VTMSYwJAYDVQQKEX10ZXRxY2FwZSBDb21tdW5pY2F0aW
9ucyBDb3JwLjEYMBYGA1UEAxMPVGltb3RoeSBBIUhvd2VzMSEwHwYJKoZIhvcNAQkBF
hJob3dlc0BuZXRzY2FwZS5jb20xFTATBgoJkiaJk/IsZAEBEwVob3dlczBcMA0GCSqG
SIb3DQEBAQUAA0sAMEgCQQC0JZf6wkG8pLMXHHCUvMfL5H6zjSk4vTTXZpYyrdN2dXc
oX49LKiOmgeJSzoIFKHtLOIboyludF90CgqcxtwKnAgMBAAGjNjA0MBEGCWCGSAGG+E
IBAQQEAwIAoDAfBgNVHSMEGDAwBT84FTob/GV3jr3mcau+hUMbsQukjANBgkqhkiG9
w0BAQQFAAOBgQBexv7o7mi3PLXadkmNP9LcIPmx93HGp0Kgyx1jIVMyNgsemeAwBM+M
SlhMfcpbTrONwNjZYW8vJDSoi//yrZlVt9bJbs7MNYZVsyFlunsqaln4/vy6Uawfg8V
UMklU7jt8LYpo4YULU7UZHPYVUaSGVttImOHZIKi4hlPXBOhcUQ==
end:vcard
```

#### 8.4. Example 4

The final example illustrates the use of the multipart/related Content-Type to include non-textual directory data via the "uri" encoding to refer to other body parts within the same message, or to external values. Note that no "profile" parameter is given, so an application may not know what kind of directory entity the information applies to. Note also the use of both hypothetical official and bilaterally agreed upon types.

```
Content-Type: multipart/related;
    boundary=woof;
    type="text/directory";
    start="<id5@host.com>"
Content-ID: <id4@host.com>

--woof
Content-Type: text/directory; charset="iso-8859-1"
Content-ID: <id5@host.com>
Content-Transfer-Encoding: Quoted-Printable

source:ldap://cn=Bjorn%20Jensen,o=University%20of%20Michigan,c=US
cn:Bj=F8rn Jensen
sn:Jensen
email:bjorn@umich.edu
image;value=uri:cid:id6@host.com
image;value=uri;format=jpeg:ftp://some.host/some/path.jpg
sound;value=uri:cid:id7@host.com
phone:+1 313 747-4454

--woof
Content-Type: image/jpeg
Content-ID: <id6@host.com>

<...image data...>

--woof
Content-Type: message/external-body;
    name="myvoice.au";
    site="myhost.com";
    access-type=ANON-FTP;
    directory="pub/myname";
    mode="image"

Content-Type: audio/basic
Content-ID: <id7@host.com>

--woof--
```

## 9. Registration of new profiles

This section defines procedures by which new profiles are registered with the IANA and made available to the Internet community. Note that non-IANA profiles can be used by bilateral agreement, provided the associated profile names follow the "X-" convention defined above.

The procedures defined here are designed to allow public comment and review of new profiles, while posing only a small impediment to the definition of new profiles.

Registration of a new profile is accomplished by the following steps.

### 9.1. Define the profile

A profile is defined by completing the following template.

```
To: ietf-mime-direct@imc.org
Subject: Registration of text/directory MIME profile XXX

Profile name:

Profile purpose:

Profile types:

Profile special notes (optional):

Intended usage: (one of COMMON, LIMITED USE or OBSOLETE)
```

The explanation of what goes in each field in the template follows.

**Profile name:** The name of the profile as it will appear in the text/directory MIME Content-Type "profile" header parameter, or the predefined "profile" type name.

**Profile purpose:** The purpose of the profile (e.g., to represent information about people, printers, documents, etc.). Give a short but clear description.

**Profile types:** The list of types associated with the profile. This list of types is to be expected but not required in the profile, unless otherwise noted in the profile definition. Other types not mentioned in the profile definition MAY also be present. Note that any new types referenced by the profile MUST be defined separately as described in [Section 10](#).

Profile special notes: Any special notes about the profile, how it is to be used, etc. This section of the template can also be used to define an ordering on the types that appear in the Content-Type, if such an ordering is required.

#### 9.2. Post the profile definition

The profile description must be posted to the new profile discussion list, [ietf-mime-direct@imc.org](mailto:ietf-mime-direct@imc.org)

#### 9.3. Allow a comment period

Discussion on the new profile must be allowed to take place on the list for a minimum of two weeks. Consensus must be reached on the profile before proceeding to step 4.

#### 9.4. Submit the profile for approval

Once the two-week comment period has elapsed, and the proposer is convinced consensus has been reached on the profile, the registration application should be submitted to the Profile Reviewer for approval. The Profile Reviewer is appointed by the Application Area Directors and can either accept or reject the profile registration. An accepted registration is passed on by the Profile Reviewer to the IANA for inclusion in the official IANA profile registry. The registration may be rejected for any of the following reasons. 1) Insufficient comment period; 2) Consensus not reached; 3) Technical deficiencies raised on the list or elsewhere have not been addressed. The Profile Reviewer's decision to reject a profile can be appealed by the proposer to the IESG, or the objections raised can be addressed by the proposer and the profile resubmitted.

### 10. Profile Change Control

Existing profiles can be changed using the same process by which they were registered.

Define the change

Post the change

Allow a comment period

Submit the changed profile for approval

Note that the original author or any other interested party can propose a change to an existing profile, but that such changes should only be proposed when there are serious omissions or errors in the published specification. The Profile Reviewer can object to a change if it is not backwards compatible, but is not required to do so.

Profile definitions can never be deleted from the IANA registry, but profiles which are no longer believed to be useful can be declared OBSOLETE by a change to their "intended use" field.

## 11. Registration of new types

This section defines procedures by which new types are registered with the IANA. Note that non-IANA types can be used by bilateral agreement, provided the associated types names follow the "X-" convention defined above.

The procedures defined here are designed to allow public comment and review of new types, while posing only a small impediment to the definition of new types.

Registration of a new type is accomplished by the following steps.

### 11.1. Define the type

A type is defined by completing the following template.

```
To: ietf-mime-direct@imc.org
Subject: Registration of text/directory MIME type XXX

Type name:

Type purpose:

Type encoding:

Type valuetype:

Type special notes (optional):

Intended usage: (one of COMMON, LIMITED USE or OBSOLETE)
```

The meaning of each field in the template is as follows.

Type name: The name of the type, as it will appear in the body of an text/directory MIME Content-Type "type: value" line to the left of the colon ":".



Type purpose: The purpose of the type (e.g., to represent a name, postal address, IP address, etc.). Give a short but clear description.

Type encoding: The default encoding a value of the type must have in the body of a text/directory MIME Content-Type.

Type valuetype: The format a value of the type must have in the body of a text/directory MIME Content-Type. This description must be precise and must not violate the general encoding rules defined in [section 5](#) of this document.

Type special notes: Any special notes about the type, how it is to be used, etc.

#### 11.2. Post the type definition

The type description must be posted to the new type discussion list, [ietf-mime-direct@imc.org](mailto:ietf-mime-direct@imc.org)

#### 11.3. Allow a comment period

Discussion on the new type must be allowed to take place on the list for a minimum of two weeks. Consensus must be reached on the type before proceeding to step 4.

#### 11.4. Submit the type for approval

Once the two-week comment period has elapsed, and the proposer is convinced consensus has been reached on the type, the registration application should be submitted to the Profile Reviewer for approval. The Profile Reviewer is appointed by the Application Area Directors and can either accept or reject the type registration. An accepted registration is passed on by the Profile Reviewer to the IANA for inclusion in the official IANA profile registry. The registration can be rejected for any of the following reasons. 1) Insufficient comment period; 2) Consensus not reached; 3) Technical deficiencies raised on the list or elsewhere have not been addressed. The Profile Reviewer's decision to reject a type can be appealed by the proposer to the IESG, or the objections raised can be addressed by the proposer and the type resubmitted.

## 12. Type Change Control

Existing types can be changed using the same process by which they were registered.

Define the change

Post the change

Allow a comment period

Submit the type for approval

Note that the original author or any other interested party can propose a change to an existing type, but that such changes should only be proposed when there are serious omissions or errors in the published specification. The Profile Reviewer can object to a change if it is not backwards compatible, but is not required to do so.

Type definitions can never be deleted from the IANA registry, but types which are no longer believed to be useful can be declared OBSOLETE by a change to their "intended use" field.

## 13. Registration of new parameters

This section defines procedures by which new parameters are registered with the IANA and made available to the Internet community. Note that non-IANA parameters can be used by bilateral agreement, provided the associated parameters names follow the "X-" convention defined above.

The procedures defined here are designed to allow public comment and review of new parameters, while posing only a small impediment to the definition of new parameters.

Registration of a new parameter is accomplished by the following steps.

### 13.1. Define the parameter

A parameter is defined by completing the following template.

To: ietf-mime-direct@imc.org

Subject: Registration of text/directory MIME type parameter XXX

Parameter name:

Parameter purpose:

Parameter values:

Parameter special notes (optional):

Intended usage: (one of COMMON, LIMITED USE or OBSOLETE)

The explanation of what goes in each field in the template follows.

Parameter name: The name of the parameter as it will appear in the text/directory MIME Content-Type.

Parameter purpose: The purpose of the parameter (e.g., to represent the format of an image, type of a phone number, etc.). Give a short but clear description. If defining a general parameter like "format" or "type" keep in mind that other applications might wish to extend its use.

Parameter values: The list or description of values associated with the parameter.

Parameter special notes: Any special notes about the parameter, how it is to be used, etc.

### 13.2. Post the parameter definition

The parameter description must be posted to the new parameter discussion list, [ietf-mime-direct@imc.org](mailto:ietf-mime-direct@imc.org)

### 13.3. Allow a comment period

Discussion on the new parameter must be allowed to take place on the list for a minimum of two weeks. Consensus must be reached on the parameter before proceeding to step 4.

### 13.4. Submit the parameter for approval

Once the two-week comment period has elapsed, and the proposer is convinced consensus has been reached on the parameter, the registration application should be submitted to the Profile Reviewer for approval. The Profile Reviewer is appointed by the Application Area Directors and can either accept or reject the parameter registration. An accepted registration is passed on by the Profile Reviewer to the IANA for inclusion in the official IANA parameter registry. The registration can be rejected for any of the following reasons. 1) Insufficient comment period; 2) Consensus not reached; 3) Technical deficiencies raised on the list or elsewhere have not been addressed. The Profile Reviewer's decision to reject a profile can be appealed by the proposer to the IESG, or the objections raised can be

addressed by the proposer and the parameter registration resubmitted.

#### 14. Parameter Change Control

Existing parameters can be changed using the same process by which they were registered.

Define the change

Post the change

Allow a comment period

Submit the parameter for approval

Note that the original author or any other interested party can propose a change to an existing parameter, but that such changes should only be proposed when there are serious omissions or errors in the published specification. The Profile Reviewer can object to a change if it is not backwards compatible, but is not required to do so.

Parameter definitions can never be deleted from the IANA registry, but parameters which are no longer believed to be useful can be declared OBSOLETE by a change to their "intended use" field.

#### 15. Registration of new value types

This section defines procedures by which new value types are registered with the IANA and made available to the Internet community. Note that non-IANA value types can be used by bilateral agreement, provided the associated value types names follow the "X-" convention defined above.

The procedures defined here are designed to allow public comment and review of new value types, while posing only a small impediment to the definition of new value types.

Registration of a new value types is accomplished by the following steps.

##### 15.1. Define the value type

A value type is defined by completing the following template.

To: ietf-mime-direct@imc.org

Subject: Registration of text/directory MIME value type XXX

value type name:

value type purpose:

value type format:

value type special notes (optional):

Intended usage: (one of COMMON, LIMITED USE or OBSOLETE)

The explanation of what goes in each field in the template follows.

value type name: The name of the value type as it will appear in the text/directory MIME Content-Type.

value type purpose: The purpose of the value type. Give a short but clear description.

value type format: The definition of the format for the value, usually using ABNF grammar.

value type special notes: Any special notes about the value type, how it is to be used, etc.

#### 15.2. Post the value type definition

The value type description must be posted to the new value type discussion list, [ietf-mime-direct@imc.org](mailto:ietf-mime-direct@imc.org)

#### 15.3. Allow a comment period

Discussion on the new value type must be allowed to take place on the list for a minimum of two weeks. Consensus must be reached before proceeding to step 4.

#### 15.4. Submit the value type for approval

Once the two-week comment period has elapsed, and the proposer is convinced consensus has been reached on the value type, the registration application should be submitted to the Profile Reviewer for approval. The Profile Reviewer is appointed by the Application Area Directors and can either accept or reject the value type registration. An accepted registration should be passed on by the Profile Reviewer to the IANA for inclusion in the official IANA value type registry. The registration can be rejected for any of the following reasons. 1) Insufficient comment period; 2) Consensus not reached; 3) Technical deficiencies raised on the list or elsewhere have not been addressed. The Profile Reviewer's decision to reject a

profile can be appealed by the proposer to the IESG, or the objections raised can be addressed by the proposer and the value type registration resubmitted.

## 16. Security Considerations

Internet mail is subject to many well known security attacks, including monitoring, replay, and forgery. Care should be taken by any directory service in allowing information to leave the scope of the service itself, where any access controls can no longer be guaranteed. Applications should also take care to display directory data in a "safe" environment (e.g., PostScript-valued types).

## 17. Acknowledgements

The registration procedures defined here were shamelessly lifted from the MIME registration RFC.

The many valuable comments contributed by members of the IETF ASID working group are gratefully acknowledged, as are the contributions of the Versit Consortium. Chris Newman was especially helpful in navigating the intricacies of ABNF lore.

## 18. References

- [RFC-1777] Yeong, W., Howes, T., and S. Kille, "Lightweight Directory Access Protocol", [RFC 1777](#), March 1995.
- [RFC-1778] Howes, T., Kille, S., Yeong, W., and C. Robbins, "The String Representation of Standard Attribute Syntaxes", [RFC 1778](#), March 1995.
- [RFC-822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, [RFC 822](#), August 1982.
- [RFC-2045] Borenstein, N., and N. Freed, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", [RFC 2045](#), November 1996.
- [RFC-2046] Moore, K., "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", [RFC 2046](#), November 1996.
- [RFC-2048] Freed, N., Klensin, J., and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", [RFC 2048](#), November 1996.
- [RFC-1766] Alvestrand, H., "Tags for the Identification of Languages", [RFC 1766](#), March 1995.

- [RFC-2112] Levinson, E., "The MIME Multipart/Related Content-type", [RFC 2112](#), March 1997.
- [X500] "Information Processing Systems - Open Systems Interconnection - The Directory: Overview of Concepts, Models and Services", ISO/IEC JTC 1/SC21, International Standard 9594-1, 1988.
- [RFC-1835] Deutsch, P., Schoultz, R., Faltstrom, P., and C. Weider, "Architecture of the WHOIS++ service", [RFC 1835](#), August 1995.
- [RFC-1738] Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)", [RFC 1738](#), December 1994.
- [MIME-VCARD] Dawson, F., and T. Howes, "VCard MIME Directory Profile", [RFC 2426](#), September 1998.
- [VCARD] Internet Mail Consortium, "vCard - The Electronic Business Card", Version 2.1, <http://www.imc.com/pdi/vcard-21.txt>, September, 1996.
- [RFC-2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC-2234] Crocker, D., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.

## 19. Authors' Addresses

Tim Howes  
Netscape Communications Corp.  
501 East Middlefield Rd.  
Mountain View, CA 94041  
USA

Phone: +1.415.937.3419  
EMail: howes@netscape.com

Mark Smith  
Netscape Communications Corp.  
501 East Middlefield Rd.  
Mountain View, CA 94041  
USA

Phone: +1.415.937.3477  
EMail: mcs@netscape.com

Frank Dawson  
Lotus Development Corporation  
6544 Battleford Drive  
Raleigh, NC 27613  
USA

Phone: +1-919-676-9515  
EMail: frank\_dawson@lotus.com



## 20. Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.