

Network Working Group  
Request for Comments: DRAFT  
[draft-ietf-asid-mime-direct-05.txt](#)

Tim Howes  
Mark Smith  
Netscape Communications Corp.  
Frank Dawson  
Lotus Development Corporation  
November 15, 1997

## **A MIME Content-Type for Directory Information**

### **1. Status of this Memo**

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this memo is unlimited.

Copyright (C) The Internet Society 1997. All Rights Reserved.

### **2. 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, must be represented.

The text/directory Content-Type defines a general framework and format for holding directory information in a simple ''type: value'' form.

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 may 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](#)].

### **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 may support creation and modification of the information as well. Directory information is usually accessed far more often than it is updated. Directories may be local or global in scope. They may be distributed or centralized. The information they contain may be replicated, with weak or strong consistency requirements.

There are several situations in which users of Internet mail may 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 may also be useful for these protocols to be able 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 parsable 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 may 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 may 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 types.

In some applications, it may 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, 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 may 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 "Person" is the same as "PERSON" and "person" and "peRsOn").

profile = x-token / iana-token

x-token = <The two characters "X-" or "x-" followed,  
with no intervening white space, by any atom,  
where atom is from [Section 3.3 of RFC 822](#)>

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. Note that each value may also have an inline encoding associated with it. This encoding is independent of the encoding for the body part as a whole. When encoding, the backslash encoding described below for text values containing special characters is applied first, inline per-value encodings are performed second, then Content-Transfer-Encoding is applied to the entire body part. When decoding, the Content-Transfer-Encoding for the entire body part is decoded, then any value-specific encodings are decoded, and finally any backslash encodings are decoded.

The backslash encoding is used to encode special characters occurring in a text value that otherwise would have special significance in the content type. The following table summarizes the special characters and their backslash encodings.

special character	backslash encoding
-----	-----
newline (ASCII decimal 10)	\n or \N
comma (ASCII decimal 44)	\,
backslash (ASCII decimal 92)	\\



## **5.6. Security considerations**

Directory information may be public or it may 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 may be protected, or by which access control information may be conveyed. Note that the integrity and privacy of a text/directory body part may 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 at any point by inserting a CRLF immediately followed by a single white space character (space, ASCII decimal 32, or horizontal tab, ASCII decimal 9). 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 description  
that exists on 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).

A formatted text line break in a 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 both the \n literal formatted line break technique and the CRLF-followed-by-space line folding technique.

Data values containing the CRLF sequence may also be encoded using the "B" encoding method, defined in [\[RFC-1522\]](#).

#### **[5.8.2.](#) BNF content-type definition**

Using the notation of [RFC 822](#), the syntax for this content is:

contentline = [group.]type [";" [ws] parameterlist] ":" [ws] valuespec

group = atom ; as defined in [Section 3.3 of RFC 822](#)

type = x-name  
/ iana-type

x-name = <the two characters "X-" or "x-" followed, with no intervening white space, by any atom>



iana-type = <a publicly-defined extension token, registered  
with IANA, as specified in [Section 11](#) of this  
document>

ws = \*(HTAB / SPACE) ; ASCII decimal 9 and 32

parameterlist = parameter \*(";" [ws] parameter)

parameter = encodingparm  
/ valuetypeparm  
/ languageparm  
/ contextparm  
/ [parmtype "="] parmvalues

encodingparm = "encoding" "=" encodingtype

encodingtype = "b" ; from [RFC 2047](#)

valuetypeparm = "value" "=" valuetype

valuetype = "uri" ; generic uri from [RFC 1738](#)  
/ "text"  
/ "date"  
/ "time"  
/ "date-time" ; date time  
/ "integer"  
/ "boolean"  
/ "float"  
/ x-name  
/ iana-valuetype

iana-valuetype : = <a publicly-defined extension token, registered  
with IANA, as defined in [Section 15](#) of this  
document>

languageparm = "language" "=" language ; as defined in [RFC 1766](#)

contextparm = "context" "=" context

context = iana-token ; a token registered with IANA  
/ x-name

parmtype = x-name  
/ iana-parmtype

iana-parmtype = <a publicly-defined extension token, registered  
with IANA, as defined in [Section 12](#) of this  
document>



parmvalues = parmvalue \*(", " parmvalue)

parmvalue = x-name  
          / iana-parmvalue

iana-parmvalue = <a publicly-defined extension token, registered  
                  with IANA, as defined in [Section 13](#) of this  
                  document, conforming to the qtext or quoted-text  
                  productions in [section 3.3 of RFC 822](#)>

valuespec = text-spec  
          / date-spec  
          / time-spec  
          / date-time-spec  
          / boolean  
          / integer  
          / float  
          / iana-valuespec

text-spec = text \*(", " [ws] text)

text = <bytes whose syntax depends on type and the  
      the encoding parameter. If the value contains  
      a CRLF sequence (ASCII 10 followed by 13), it must  
      be encoded using either the "b" or \n methods.>

date-spec = date \*(", " [ws] date)

time-spec = time \*(", " [ws] time)

date-time-spec = date "T" time

boolean = "TRUE" / "FALSE"

integer = [sign] 1\*digit \*(", " [ws] [sign] 1\*digit)

float = [sign] 1\*digit ["." \*digit] \*(", " float)

sign = "+" / "-"

digit = "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"

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

date-fullyear = 4digit

date-month = 2digit       ;01-12



```
date-mday = 2digit      ;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 = 2digit      ;00-23

time-minute = 2digit    ;00-59

time-second = 2digit    ;00-59

time-secfrac = "," 1*digit

time-zone = "Z" / time-numzome

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

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

White space characters (namely HTAB and SPACE characters, ASCII decimal [9](#) and 32) may follow the SEMICOLON ';' parameter separator, the COMMA ',' list value separator or the COLON ':' value separator. Note that this means that if a "valuespec" begins with white space, it must be encoded using either the "B" encoding method.

A line that begins with a white space character is a continuation of the previous line, as described above. The CRLF and immediately following white space character 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><SPACE> or <CRLF><HTAB> found anywhere in the content indicates a continued line and should be removed.

The meanings of the various type names and the format of the corresponding values must be 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.

Type names and parameter names are case insensitive (e.g., the type name "cn" is the same as "CN" and "Cn"). 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 MAY support encoding multiple values in a single content line, for example by separating the values with a comma "," or other delimiter. This approach has been taken for several of the content types defined above (date, time, integer, float), for space-saving reasons.

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 MAY be 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 "name" type, defined below.

The "encoding" type parameter is used to specify an alternate encoding for a value. If the value contains a CRLF sequence (ASCII 10 followed by 13), it must be encoded, since CRLF is used to separate lines in the content-type itself. The CRLF sequence must be encoded 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".

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 may be split across multiple physical lines in the content type by using the <CRLF><SPACE | HTAB> line folding technique.

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.



Each type has associated with it a default encoding, taken from the Content-Transfer-Encoding MIME header parameter.

The "value" parameter is optional, and MAY be 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 may 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 may still be interpreted.

### **5.8.3. Predefined value formats**

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.

"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](#).

"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 may 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 32-bit signed integers in decimal format. The valid range for "integer" is -2147483648 to 2147483647. If sign is not specified, the value is assumed positive "+". Multiple "integer" values may be specified using the comma-separated notation, unless restricted by a profile.

Examples: 1234567890  
-123456790  
+123456790, 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 may 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  
USA  
fdawson@earthlink.net  
+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: text containing a URI.

Type special notes: The SOURCE type is used to provide the means by which applications knowledgeable in the given directory service protocol may 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 may pick what it considers to be the best source, or multiple SOURCE types may be included. The interpretation of the value for a SOURCE type may depend on the setting of the CONTEXT type parameter.

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: 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 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: person

#### **[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 delimit 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 may 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 may contain multiple entities within the same text/directory content-type or to define content that may 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 delimit 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 may 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 may contain multiple entities within the same text/directory content-type or to define content that may 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=3Dbjorn%20Jensen, o=3Duniversity%20of%20Michigan, c=3DUS
name: Bjorn Jensen
fn: Bj=F8rn Jensen
n: Jensen;Bj=F8rn
email;type=3Dinternet: bjorn@umich.edu
tel;type=3Dwork,voice,msg:+1 313 747-4454
key;type=3Dx509;encoding=3DB: 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.

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

```
source: ldap://cn=Meister%20Berger,o=Universitaet%20Goerlitz,c=DE
name: Meister Berger
cn: Meister Berger
cn: Berger Meister
sn: Berger
age;value=integer: 33
o: Universit=E6t G=F6rlitz
title: Mayor
title;language=de;value=text: Burgermeister
description: The Mayor of the great city of
  Goerlitz in the great country of Germany.
email: mb@goerlitz.de
home.phone;fax,voice,msg: +49 3581 123456
home.addr: Hufenshlage1 1234\n
  02828 Goerlitz\n
  Deutschland
certificate; encoding=b:
```

```
MIICajCCAdOgAwIBAgICBEUwDQYJKoZIhvcNAQEEBQAwZELMAkG
```

```
A1UEBhMCVVMxLDAqBgNVBAoTIO5ldHNjYXB1IENvbW11bm1jYXRpb25zIENvcnBvcnF0aW9uMRww
```

```
GgYDVQQLEXNJbmZvcn1hdGlvb1BTeXN0ZW1zMRwwGgYDVQQDEXNyY290Y2EubmV0c2NhcnGUuY29t
```

```
MB4XDTk3MDYwNjE5NDc1OVVoXDTk3MTIwMzE5NDc1OVowgYkxCzAJBgNVBAYTA1VTMSYwJAYDVQQK
```

```
Ex10ZXRzY2FwZSBDb21tdW5pY2F0aW9ucyBDb3JwLjEYMBYGA1UEAxMPVGltb3RoeSBBIHvd2Vz
  MSEWhWYJKoZIhvcNAQkBFhJob3dlc0BuZXRzY2FwZS5jb20xFTATBgoJkiaJk/
  IsZAEBEwVob3dl
```

```
czBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQC0JZf6wkg8pLMXHHCUvMfL5H6zjSk4vTTXZpYyrdN2
```

```
dXcoX49LKioMgeJSzoiFKHtLOIboyludF90CgqcxtwKnAgMBAAGjNjA0MBEGCWCGSAGG+EIBAQQE
  AwIAoDAfBgNVHSMEGDAWgBT84FT0B/
  GV3jr3mcau+hUMbsQukjANBgkqhkiG9w0BAQQFAA0BgQBe
```

```
xv7o7mi3PLXadkmNP9LcIPmx93HGp0KgYx1jIVMyNgsemeAwBM+MSlhmfcpbTrONwNjZYW8vJDSO
  i//yrZlvt9bJbs7MNYZVsyF1unsqaln4/
  vy6Uawfg8VUMk1U7jt8LYpo4YULU7UZHPYVUaSgVttI
```

m0HZIKi4hlPXBOhcUQ==

#### **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.

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=3DBjorn%20Jensen,o=3DUniversity%20of%20Michigan,c=3DUS
cn: Bj=F8rn Jensen
sn: Jensen
email: bjorn@umich.edu
image;encoding=3DUri: cid:id6@host.com
image;encoding=3DUri;format=3Djpeg: ftp://some.host/some/path.jpg
sound;encoding=3DUri: 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 may 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 may 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



### **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 to the Application Area Directors and may either accept or reject the profile registration. An accepted registration should be 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 may 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 may 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 may 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 may 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 may 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 to the Application Area Directors and may either accept or reject the type registration. An accepted registration should be 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 type may 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 may 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 may 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 may 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 may 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 may 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 to the Application Area Directors and may either accept or reject the parameter registration. An accepted registration should be passed on by the Profile Reviewer to the IANA for inclusion in the official IANA parameter 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 may 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 may 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 may 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 may 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 may 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 BNF 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 to the Application Area Directors and may 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 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 may 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..



## **18. References**

- [RFC-1777] Yeong, W., Howes, T., Kille, S., "Lightweight Directory Access Protocol", Request for Comment (RFC) 1777, March 1995.
- [RFC-1778] Howes, T., Kille, S., Yeong, W., Robbins, C.J., "The String Representation of Standard Attribute Syntaxes", Request for Comment (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., Freed, N., "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., Postel, J., "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., Weider, C., "Architecture of the WHOIS++ service", August 1995.
- [RFC-1738] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform Resource Locators (URL)", [RFC 1738](#), December 1994.
- [DATETIME] Newman, C., "Date and Time on the Internet", Internet Draft [draft-newman-datetime-01.txt](#), January 1997.
- [MIME-VCARD] F. Dawson, T. Howes, "VCard MIME Directory Profile", Internet-Draft [draft-ietf-asid-mime-vcard-04.txt](#), November, 1997.
- [VCARD] Versit Consortium, "vCard - The Electronic Business Card",



Version 2.1, <http://www.imc.com/pdi/vcard-21.txt>, September, 1996.

[RFC-2119] "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 1997.

## **19. Author's Address**

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  
USA  
fdawson@earthlink.net  
+1-919-676-9515

## **20. Table of Contents**



