Network Working Group
Requests for Comments 1278

S.E. Hardcastle-Kille
University College London

November 1991

A string encoding of Presentation Address

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard. Distribution of this memo is unlimited.

Abstract

There are a number of environments where a simple string encoding of Presentation Address is desirable. This specification defines such a representation.

1 Introduction

OSI Application Entities use presentation addresses to address other Application Entities. The model for this is defined in [ISO87b]. Presentation addresses are stored in the OSI Directory using an ASN.1 representation defined by the OSI Directory [CCI88]. Logically, a presentation address consists of:

- A presentation selector
- A session selector
- A transport selector
- A set of network addresses

The selectors are all octet strings, but often have IA5 character representations. The format of network addresses is defined in [ISO87a].

There is a need to represent presentation addresses as strings in a number of different contexts. This Internet Draft defines a format for use on the Internet. It is for display to human users, and its use is recommended whenever this needs to be done. Typically, this will be for system managers rather than for end users. It is not intended for internal storage.

This Internet Draft was originally published as UCL Research Note RN/89/14 [Kil89]. It was agreed as a unified syntax for the THORN and ISODE projects. It is used throughout ISODE.

Christian Huitema of Inria and Marshall Rose of PSI Inc. gave much useful input to this document.

2 Requirements

The main requirements are:

- Must be able to specify any legal value.
- Should be clean in the common case of the presentation address containing network addresses and no selectors.
- Must deal with selectors in the following encodings:
  - IA5
  - Decimal digits encoded as IA5 (this is the most common syntax in Europe, as it is required by X.400(84) and should receive a straightforward encoding)
  - Numeric encoded as a 16 bit unsigned integer (US GOSIP). This is mapped onto two octets, with the first octet being the high order byte of the integer.

    - General Hexadecimal

- Should give special encodings for the ad hoc encoding proposed in “An interim approach to use of Network Addresses” [HK91].
    - X.25(80) Networks
    - TCP/IP Networks
- Should be extensible for additional forms.
- Should provide a reasonably compact representation .

3 Format

The BNF is given in figure 1.

```
<digit> ::= [0-9]
<other> ::= [0-9a-zA-Z+-.]
<domainchar> ::= [0-9a-zA-Z-.]
<hexdigit> ::= [0-9a-fA-F]
<hexoctet> ::= <hexdigit> <hexdigit>
<decimaloctet> ::= <digit> | <digit> <digit>
                           | <digit> <digit> <digit>

<digitstring> ::= <digit> <digitstring>
                | <digit>
<otherstring> ::= <other> <otherstring>
                | <other>
<domainstring> ::= <domainchar> <otherstring>
                | <domainchar>
<hexstring> ::= <hexoctet> <hexstring> | <hexoctet>

<dotstring> ::= <decimaloctet> "." <dotstring>
                  | <decimaloctet> "." <decimaloctet>


<dothexstring> ::= <dotstring> | <hexstring>


<presentation-address> ::=
          [[[ <psel> "/" ] <ssel> "/" ] <tsel> "/" ]
          <network-address-list>

<network-address-list> ::= <network-address> "|" <network-address-list>
                  | <network-address>

<psel> ::= <selector>
<ssel> ::= <selector>
<tsel> ::= <selector>

<selector> ::= '"' <otherstring> '"'      -- IA5
                                -- For chars not in this
```

```
                                -- string use hex
          | "#" <digitstring>       -- US GOSIP
          | "'" <hexstring> "'H"      -- Hex
          | ""                      -- Empty but present

<network-address> ::=   "NS" "+" <dothexstring>
                        -- Concrete Binary Representation
                        -- This is the compact encoding
     | <afi> "+" <idi> [ "+" <dsp> ]
                                      -- A user oriented form
         | <idp> "+" <hexstring>
                                      -- ISO 8348 Compatability

<idp> ::= <digitstring> -

<dsp>  ::=
     | "d" <digitstring>          -- Abstract Decimal
     | "x" <dothexstring>           -- Abstract Binary
     | "l" <otherstring>           -- IA5: local form only
     | "RFC-1006" "+" <prefix> "+" <ip>
       [ "+" <port> [ "+" <tset> ]]
     | "X.25(80)" "+" <prefix> "+" <dte>
       [ "+" <cudf-or-pid> "+" <hexstring> ]
     | "ECMA-117-Binary" "+" <hexstring> "+" <hexstring>
       "+" <hexstring>
     | "ECMA-117-Decimal" "+" <digitstring> "+"
       <digitstring> "+" <digitstring>

<idi> ::= <digitstring>
<afi> ::= "X121" | "DCC" | "TELEX" | "PSTN" | "ISDN"
          | "ICD" | "LOCAL"

<prefix> ::= <digit> <digit>

<ip> ::= <domainstring>
                  -- dotted decimal form (e.g., 10.0.0.6)
                  -- or domain (e.g., twg.com)
<port> ::= <digitstring>
<tset> ::= <digitstring>

<dte> ::= <digitstring>
<cudf-or-pid> ::= "CUDF" | "PID"
```

Figure 1: String BNF

Four examples:

`"256"/NS+a433bb93c1|NS+aa3106`

`#63/#41/#12/X121+234219200300`

`'3a'H/TELEX+00728722+X.25(80)+02+00002340555+CUDF+"892796"`

`TELEX+00728722+RFC-1006+03+10.0.0.6`

Note that the RFC 1006 encoding permits use of either a DNS Domain Name or an IP address. The former is primarily for ease of entry. If this DNS Domain Name maps onto multiple IP addresses, then multiple network addresses should be generated. The DNS Domain Name form is for convenient input. When mapping from an encoded address to string form, the IP address form should always be used.

4 Encoding

Selectors are represented in a manner which can be easily encoded. In the NS notation, the concrete binary form of network address is given. Otherwise, this string notation provides a mechanism for representing the Abstract Syntax of a Network Address. This must be encoded according to Addendum 2 of ISO 8348 [ISO87a].

5 Macros

There are often common addresses, for which a cleaner representation is desired. This is achieved by use of Macros. If a `<network-address>` can be parsed as:

```
<otherstring> "=" *( any )
```

Then the leading string is taken as a Macro, which is substituted. This may be applied recursively. When presenting Network Address to humans, the longest available substitution should be used. For example:

Macro	Value
UK.AC	DCC+826+d110000
Leeds	UK.AC=120

Then "Leeds=22" would be expanded to "DCC+826+d11000012022".

6 Standard Macros

No Macros should ever be relied on. However, the following are suggested as standard.

Macro	Value
Int-X25(80)	TELEX+00728722+X25(80)+01+
Janet-X25(80)	TELEX+00728722+X25(80)+02+
Internet-RFC-1006	TELEX+00728722+RFC-1006+03+
IXI	TELEX+00728722+RFC-1006+06+

7 References

References

[CCI88] The Directory — overview of concepts, models and services, December 1988. CCITT X.500 Series Recommendations.

[HK91] S.E. Hardcastle-Kille. Encoding network addresses to support operation over non-osi lower layers. Request for Comments RFC 1277, Department of Computer Science, University College London, November 1991.

[ISO87a] Information processing systems - data communications - network services definition: Addendum 2 - network layer addressing, March 1987. ISO TC 97/SC 6.

[ISO87b] ISO DIS 7498-3 on naming and addressing, May 1987. ISO/IEC/JTC-1/SC 21.

[Kil89] S.E. Kille. A string encoding of presentation address. Research Note RN/89/14, Department of Computer Science, University College London, February 1989.

8 Security Considerations

Security considerations are not discussed in this memo.

9 Author's Address

Steve Hardcastle-Kille
Department of Computer Science
University College London
Gower Street
WC1E 6BT
England

Phone: +44-71-380-7294

EMail: S.Kille@CS.UCL.AC.UK