

# RFC 4001 : Textual Conventions for Internet Network Addresses

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 9 mars 2009

Date de publication du RFC : Février 2005

<https://www.bortzmeyer.org/4001.html>

---

Il y a plein de normes Internet dans lesquelles il faut représenter sous forme texte <<https://www.bortzmeyer.org/representation-texte.html>> des adresses IP. Parmi elles, les MIB, normalisées dans le RFC 2578<sup>1</sup> pour la gestion des réseaux (cf. section 2 du RFC et le RFC 3410). Comment représenter une adresse ? La question est un peu plus difficile que cela ne parait et méritait une norme.

Notre RFC spécifie donc à quoi doit ressembler une adresse IP dans une MIB. Par exemple, le RFC 4292 normalise une MIB pour le routage des paquets IP et doit donc indiquer des adresses, par exemple pour les objets `inetCidrRouteNextHop` qui indiquent l'adresse du « routeur suivant ». Notez que notre RFC 4001 peut même être utilisé en dehors du monde des MIB comme dans le RFC 5388 qui décrit un schéma XML pour les résultats d'un traceroute et qui doit donc également contenir des adresses IP.

Contrairement à une croyance répandue, il n'existe pas de norme générale pour la représentation des adresses IPv4. Le RFC 791 est muet à ce sujet et, bien qu'il y ait une représentation courante (192.0.2.134), d'autres représentations sont possibles <<https://www.bortzmeyer.org/representation-texte.html>>. Pour IPv6, en revanche, il existe une représentation texte normalisée, dans le RFC 5952.

Notez bien que le RFC 4001 ne normalise que la représentation d'adresses IP « pures » sans les informations liées à la couche transport comme le port. Pour des adresses généralisées, voir le RFC 3419.

C'est la section 3 qui contient les définitions des représentations des adresses IP en ASN.1. On note que les noms de domaine sont inclus comme alternative :

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc2578.txt>

```

InetAddressType ::= TEXTUAL-CONVENTION
...
    ipv4(1)      An IPv4 address as defined by the
                  InetAddressIPv4 textual convention.

    ipv6(2)      An IPv6 address as defined by the
                  InetAddressIPv6 textual convention.
...
    dns(16)     A DNS domain name as defined by the
                  InetAddressDNS textual convention.
...
InetAddressIPv4 ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "1d.1d.1d.1d"
    STATUS       current
    DESCRIPTION
        "Represents an IPv4 network address:
         Octets  Contents          Encoding
          1-4   IPv4 address      network-byte order
...
InetAddressIPv6 ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "2x:2x:2x:2x:2x:2x:2x:2x"
    STATUS       current
    DESCRIPTION
        "Represents an IPv6 network address:

         Octets  Contents          Encoding
          1-16   IPv6 address      network-byte order

        The corresponding InetAddressType value is ipv6(2).
...
    SYNTAX      OCTET STRING (SIZE (16))

```

La section 4, quant à elle, donnant des indications sur l'usage de ces variables. On trouve aussi des adresses munies d'un "zone index", qui indique le réseau auquel elles appartiennent, pour le cas d'adresses locales au lien par exemple celles du RFC 3927 pour IPv4 ou du RFC 4007 pour IPv6 (section 4.2).

Enfin, le RFC se conclut sur un exemple d'utilisation (section 5) pour une MIB imaginaire de `peerAddress` :

```

peerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PeerEntry
    ...
    ::= { somewhere 1 }

peerEntry OBJECT-TYPE
    SYNTAX      PeerEntry
    ...
    INDEX       { peerAddressType, peerAddress }
    ::= { peerTable 1 }

PeerEntry ::= SEQUENCE {
    peerAddressType  InetAddressType,
    peerAddress      InetAddress,
    ...
}

peerAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    ...
    ::= { peerEntry 1 }

peerAddress OBJECT-TYPE
    SYNTAX      InetAddress (SIZE (1..64))
    ...

```

## DESCRIPTION

```
"The Internet address for the peer. The type of this
address is determined by the value of the peerAddressType
object. ...
::= { peerEntry 2 }
```

Ce RFC remplace le RFC 3291 avec peu de changements (détaillés en section 8).