

Ethereum nodes have to identify themselves with some basic information to connect to peers. To ensure any potential peer can interpret this information, it is relayed in one of three standardized formats that any Ethereum node can understand: multiaddr, enode, or Ethereum Node Records (ENRs). ENRs are the current standard for Ethereum network addresses.

Prerequisites {#prerequisites}

Some understanding of Ethereum's [networking layer](#) is required to understand this page.

Multiaddr {#multiaddr}

The original Ethereum node address format was the 'multiaddr' (short for 'multi-addresses'). Multiaddr is a universal format designed for peer-to-peer networks. Addresses are represented as key-value pairs with keys and values separated with a forward slash. For example, the multiaddr for a node with IPv4 address 192.168.22.27 listening to TCP port 33000 looks like:

```
/ip4/192.168.22.27/tcp/33000
```

For an Ethereum node, the multiaddr contains the node-ID (a hash of their public key):

```
/ip4/192.168.22.27/tcp/33000/p2p/5t7Nv7dG2d6fFbvAiewVsEwWweU3LdebSqX2y1bPrW8br
```

Enode {#enode}

An enode is a way to identify an Ethereum node using a URL address format. The hexadecimal node-ID is encoded in the username portion of the URL separated from the host using an @ sign. The hostname can only be given as an IP address; DNS names are not allowed. The port in the hostname section is the TCP listening port. If the TCP and UDP (discovery) ports differ, the UDP port is specified as a query parameter "discport"

In the following example, the node URL describes a node with IP address 10.3.58.6, TCP port 30303 and UDP discovery port 30301.

```
enode://6f8a80d14311c39f35f516fa664deaaaa13e85b2f7493f37f6144d86991ec012937307647bd3b9a82abe2974e1407241d54947bbb39763a4cac9f77166ad92a0@10.3.58.6:30303?discport=30301
```

Ethereum Node Records (ENRs) {#enr}

Ethereum Node Records (ENRs) are a standardized format for network addresses on Ethereum. They supersede multiaddr's and enodes. These are especially useful because they allow greater informational exchange between nodes. The ENR contains a signature, sequence number and fields detailing the identity scheme used to generate and validate signatures. The ENR can also be populated with arbitrary data organized as key-value pairs. These key-value pairs contain the node's IP address and information about the sub-protocols the node is able to use. Consensus clients use a [specific ENR structure](#) to identify boot nodes and also include an eth2 field containing information about the current Ethereum fork and the attestation gossip subnet (this connects the node to a particular set of peers whose attestations are aggregated together).

Further Reading {#further-reading}

[EIP-778: Ethereum Node Records \(ENR\) Network addresses in Ethereum LibP2P: Multiaddr-Enode-ENR?](#)