

Ethereum Improvement Proposals

[All](#) [Core](#) [Networking](#) [Interface](#) [ERC](#) [Informational](#) [Meta](#)

EIP 634: Storage of text records in ENS «»

Author	Richard Moore
Status	Draft
Type	Standards Track
Category	ERC
Created	2017-05-17

Abstract

This EIP defines a resolver profile for ENS that permits the lookup of arbitrary key-value text data. This allows ENS name holders to associate e-mail addresses, URLs and other informational data with a ENS name.

Motivation

There is often a desire for human-readable metadata to be associated with otherwise machine-driven data; used for debugging, maintenance, reporting and general information.

In this EIP we define a simple resolver profile for ENS that permits ENS names to associate arbitrary key-value text.

Specification

Resolver Profile

A new resolver interface is defined, consisting of the following method:

```
function text(bytes32 node, string key) constant returns (string text);
```

The interface ID of this interface is 0x59d1d43c.

The `text` data may be any arbitrary UTF-8 string. If the key is not present, the empty string must be returned.

Initial Recommended Keys

Keys must be made up of lowercase letters, numbers and the hyphen (-). Vendor specific services must be prefixed with `vnd.`.

- **email** - an e-mail address
- **url** - a URL
- **avatar** - a URL to an image used as an avatar or logo
- **description** - A description of the name
- **notice** - A notice regarding this name;
- **keywords** - A list of comma-separated keywords, ordered by most significant first; clients that interpret this field may choose a threshold beyond which to ignore
- **vnd.twitter** - a twitter username (should it be prefixed with an @?)
- **vnd.github** - a GitHub username

Rationale

Application-specific vs general-purpose record types

Rather than define a large number of specific record types (each for generally human-readable data) such as `url` and `email`, we follow an adapted model of DNS's `TEXT` records, which allow for a general keys and values, allowing future extension without adjusting the resolver, while allowing applications to use custom keys for their own purposes.

Backwards Compatibility

Not applicable.

Test Cases

TBD

Implementation

None yet.

Copyright

Copyright and related rights waived via [CC0](#).

Ethereum Improvement Proposals

Ethereum Improvement Proposals



[ethereum/EIPs](#)



[ethereum](#)

Ethereum Improvement Proposals (EIPs) describe standards for the Ethereum platform, including core protocol specifications, client APIs, and contract standards.