

.eth Permanent Registrar

The Permanent Registrar is the code that will govern allocation and renewal of names in the .eth TLD. Presently this is governed by the legacy auction registrar, which uses a Vickery Auction process to allocate names to registrants. The new registrar aims to simplify this process, while providing a stable platform for future improvements that will minimise API changes.

The target deployment date for the permanent registrar is the 4th of May, 2019. Documentation provided here is preliminary, and intended to provide developers wanting to integrate .eth domain registration or renewal into their platforms or tools with a starting point.

System architecture

Code for the permanent registrar can be found in the [ethregistrar](#) repository.

The registrar itself is called [BaseRegistrar](#). This contract implements several key functions:

- The owner of the registrar may add and remove 'controllers'.
- Controllers may register new domains and extend the expiry of (renew) existing domains. They can not change the ownership or reduce the expiration time of existing domains.
- Name owners may transfer ownership to another address.
- Name owners may reclaim ownership in the ENS registry if they have lost it.
- Owners of names in the legacy registrar may transfer them to the new registrar, during the 1 year transition period. When they do so, their deposit is returned to them in its entirety.

In addition, the registrar is an [ERC721](#) compliant nonfungable token contract, meaning that .eth registrations can be transferred in the same fashion as other NFTs.

Users will interact directly with this contract when transferring ownership of names, or recovering ownership in the ENS registry of a name (for example, one whose ownership was previously

transferred to a contract). Users can also query names to see their registration status and expiry date. For initial registration and for renewals, users will need to interact with a controller contract.

This separation of concerns reduces the attack surface for the registrar, and provides users with guarantees of continued ownership of a name so long as the registrar is in place. Simultaneously, it provides for improvement and innovation over registration and renewal mechanisms. A future update may transfer ownership of the root and the .eth TLD to a contract with restricted permissions, thus preventing even the root keyholders from modifying a .eth registraion, while still providing for future updates to the set of controllers.

Initially, one controller is implemented, the [ETHRegistrarController](#). This controller provides a straightforward registration and renewal mechanism for domains that are 7 or more characters long, implementing the following functionality:

- The owner of the controller may set a price oracle contract, which determines the cost of registrations and renewals based on the name and the desired registration or renewal duration.
- The owner of the controller may withdraw any collected funds to their account.
- Users can register new names using a commit/reveal process and by paying the appropriate registration fee.
- Users can renew a name by paying the appropriate fee. Any user may renew a domain, not just the name's owner. There is no limit on renewal duration.

By allowing anyone to renew a domain, users concerned with the longevity of a name they interact with can ensure it remains registered by paying for the registration themselves, if necessary.

By allowing renewal for arbitrarily long periods of time, users can 'lock in' a desirable registration fee. Names can be made effectively 'immortal' by renewing for a long period, ensuring that stability of the name can be guaranteed by smart contract.

Users will interact with this controller for registering domains 7+ characters long, and for renewing any domain. After the restriction on name length has been relaxed and an auction has been conducted for initial allocation of shorter names, a revised version of this controller will be deployed, allowing registration of shorter names as well.

Initially, a single pricing oracle will be deployed, the [StablePriceOracle](#). This contract permits its owner to set prices in USD for each permitted name length, and uses a USD:ETH price oracle to convert those prices into Ether at the current rate. Users will not have to interact with this oracle

directly, as the controller provides functionality to determine pricing for a candidate name registration or renewal.

Discovery

Finding the address of the new registrar is straightforward: look up the owner of the domain 'eth' in ENS, by calling `owner(namehash('eth'))` on the ENS registry.

In order to support discovering the address of the controller, ENS supports interface discovery via [EIP 1844](#). This mechanism permits looking up the address of the contract that implements a required interface via the following process:

1. Set `node = namehash('eth')` .
2. Look up the ENS resolver by calling `resolver(node)` on the ENS registry.
3. Call the `interfaceImplementer(node, interfaceId)` method on that resolver, where `interfaceId` is the [EIP 165](#) interface ID of the interface you need.

The following interface IDs are presently defined for the .eth permanent registrar:

- `0x6ccb2df4` , the interface ID for [ERC721](#) (NFTs). This returns the address of the registrar itself (which can also be fetched by doing an address lookup, or by looking up the owner of 'eth').
- `0x018fac06` , the interface ID for the controller. Returns the controller's address.
- `0x7ba18ba1` , the interface ID for the legacy registrar's migration function. Returns the legacy registrar's address.