

## EIP

**Scope:** These rules shall apply to this contract.

**Underlying:** The Underlying for this Contract is the status of Ethereum Improvement Proposals and Protocol -<count> (EIPs) after Issuance and before <date>. Revisions to the Underlying made after Expiration will not be accounted for in determining the Expiration Value.

**Instructions:** The Underlying can be found at: <https://github.com/ethereum/EIPs>. These instructions on how to access the Underlying are provided for convenience only and are not part of the binding Terms and Conditions of the Contract. They may be clarified at any time.

**Source Agency:** The Source Agencies are the Ethereum network, the EIP GitHub Repository and the Ethereum foundation.

**Type:** The type of Contract is an Event Contract.

**Issuance:** After the initial Contract, Contract iterations will be listed on an as-needed basis at the discretion of the Exchange and corresponding to the risk management needs of Members.

**<count>:** The Exchange may list iterations of the Contract with <count> levels that fall within an inclusive range between 0 and 1000000000 at consecutive increments of <1>. Due to the potential for variability in the Underlying, the Exchange may modify <count> levels in response to suggestions by Members.

**<date>:** <date> refers to a calendar date specified by the Exchange. The Exchange may list iterations of the Contract corresponding to variations of <date>. For the purposes of this Contract, <date> will be considered to end starting with 12:00 AM ET the next day.

**Payout Criterion:** The Payout Criterion for the Contract encompasses the Expiration Values that EIP-<count> has become effective after Issuance and before <date>. For the purposes of this market, “effective” means that:

- For core proposals, the proposal must be implemented and activated on the Ethereum mainnet after Issuance and before <date>.
- For non-core proposals, the proposal must be implemented and activated on the Ethereum mainnet after Issuance and before <date> or have reached “Final” status on the EIP repository (or, for Meta and Informational EIPs that are not meant to be Final like continuous policy documents, Active/Living status).
- Implementations on test networks or side-chains do not count towards the EIP being “effective.”
- Milestones like Draft, Review, Last Call, or Accepted statuses, or inclusion in a scheduled upgrade, do not by themselves make the EIP effective for this market – only the Final status (or activation on mainnet) by the deadline counts.

In the event of a hard fork that results in multiple chains, ‘Ethereum mainnet’ refers to the chain that the Ethereum Foundation continues to support (i.e., the chain retaining the ticker ETH on major exchanges).

Only changes to an EIP’s status that are merged into the official EIP repository by an authorized editor, or officially announced as part of an Ethereum network upgrade by the Foundation, will be recognized.

**Examples:**

- **Example 1 (Core EIP – Yes Outcome):** EIP-1559 was scheduled for inclusion in the London hard fork. The London upgrade was executed on August 5, 2021, activating EIP-1559 on Ethereum mainnet. If a contract asked “Will EIP-1559 become effective by August 31, 2021?”, it would resolve to Yes, since the proposal was implemented and its status became Final in early August 2021 (well before the deadline).
- **Example 2 (Core EIP – No Outcome):** Suppose EIP-12345 is accepted by developers and slated for a future fork, but by the target date the upgrade hasn’t happened. The EIP remains in non-Final status. The contract “Will EIP-12345 become effective by [Date]?” would resolve No because the mainnet activation did not occur in time (and the EIP was not Final by the deadline).
- **Example 3 (ERC Standard – Yes Outcome):** EIP-20 (the ERC-20 token standard) went through the EIP process and was marked Final on November 19, 2015 (hypothetically). A contract on whether “EIP-20 becomes effective by end of 2015” would resolve Yes, as the proposal reached Final status (indicating it’s an approved standard) by the end of that year.
- **Example 4 (Standard EIP – No Outcome):** EIP-99999 is an Ethereum standard proposal in Draft as of mid-year. If the contract’s deadline passes and EIP-99999 is still not Final (perhaps it’s in Last Call or still under discussion), then it hasn’t become effective. The contract resolves No.
- **Example 5 (Superseded EIP – No Outcome):** EIP-30000 is a draft proposal that later gets merged into EIP-30500 and EIP-30000 itself is withdrawn. A contract on EIP-30000 would resolve No (not effective), even if EIP-30500 eventually became Final, because the specific EIP in question (30000) never independently achieved effective status.

**Minimum Tick:** The Minimum Tick size for the Contract shall be \$0.01.

**Position Accountability Level:** The Position Accountability Level for the Contract shall be \$25,000 per strike, per Member.

**Last Trading Date:** The Last Trading Date of the Contract will be the day prior to <date>. The Last Trading Time will be 11:59 PM ET.

**Settlement Date:** The Settlement Date of the Contract shall be no later than the day after the Expiration Date, unless the Market Outcome is under review pursuant to Rule 7.1.

**Expiration Date:** The latest Expiration Date of the Contract shall be one week after <date>. If an event described in the Payout Criterion occurs, expiration will be moved to an earlier date and time in accordance with Rule 7.2.

**Expiration time:** The Expiration time of the Contract shall be 10:00 AM ET.

**Settlement Value:** The Settlement Value for this Contract is \$1.00.

**Expiration Value:** The Expiration Value is the value of the Underlying as documented by the Source Agency on the Expiration Date at the Expiration time.

**Contingencies:** Before Settlement, Kalshi may, at its sole discretion, initiate the Market Outcome Review Process pursuant to Rule 6.3(d) of the Rulebook. If an Expiration Value cannot be determined on the Expiration Date, Kalshi has the right to determine payouts pursuant to Rule 6.3(b) in the Rulebook.

