Governance Functionality

Below is a current list of all module parameters thatx/gov has the ability to update directly. Further documentation will be released which outlines overviews of each custom module, how modules interact with one another, and technical guides regarding how to properly submit governance proposals.

Trading Stats & Fees

Stats Module

The Stats Module tracks user maker and taker volumes over a period of time (aka look-back window). This is currently set to 30 days. The maker and taker volume info is used to place users in corresponding fee-tiers.

Governance has the ability to update the params of the Stats Module, which defines the look-back window (measured in seconds).Proto(opens in a new tab)

FeeTiers Module

Governance has the ability to update fee tiers (proto(opens in a new tab)). To read more about fee tiers head toV4 Deep Dive: Rewards and Parameters(opens in a new tab) .

Trading Core

Insurance Fund

Governance has the ability to send funds from the Protocol's Insurance Fund. Funds can be sent to individual accounts, or other modules.

Note: any account has the ability to send assets to the Insurance Fund.

Liquidations Config

Governance has the ability to adjust how liquidations are processedProto(opens in a new tab)

- Max Insurance Fund quantums for deleveraging: The maximum number of quote quantums (exclusive) that the insurance fund can have for deleverages to be enabled.
- The maximum liquidation fee, in parts-per-million. 100% of this fee goes to the Insurance Fund
- The maximum amount of how much a single position can be liquidated within one block.
- The maximum amount of how much a single subaccount can be liquidated within a single block
- Fillable price config: configuration regarding how the fillable-price spread from the oracle price increases based on the adjusted bankruptcy rating of the subaccount.

Funding Rate

Governance has the ability to adjust Funding Rate parameters:

- Funding rate clamp factor, premium vote clamp factor, and min number of votes per premium sampleProto(opens in a new tab)
- Epoch information, which defines the funding interval and premium sampling intervalProto(opens in a new tab)
- Liquidity Tier, which defines the impact notional value.Proto(opens in a new tab)

Trading Rewards

Vest Module

The Vest Module is responsible for determining the rate of tokens that vest from Vester Accounts to other accounts such as a Community Treasury Account and a Rewards Treasury Account. The rate of token transfers is linear with respect to time. Thus, block timestamps are used to vest tokens.

Governance has the ability to create, update, or delete aVestEntry (proto(opens in a new tab)), which defines:

- The start and end time of vesting
- The token that is vested
- The account to vest tokens to
- The account to vest tokens from

Rewards Module

The Rewards Module distributes trading rewards to traders (previously written about in V4 Deep Dive: Rewards and Parameters(opens in a new tab)). Governance has the ability to adjust the following (proto(opens in a new tab)):

- Which account Trading Rewards are funded from
- The token Trading Rewards are funded in
- The market which tracks the oracle price of the token that Trading Rewards are funded in
- C
- which is a protocol constant further explained in the post linked above

Markets

Oracles

Governance has the ability to adjust the list of oracles used for each marketProto(opens in a new tab)

Note that this functionality does not include creating / removing an exchange-source supported by the protocol as a whole, which will require a binary upgrade.

Liquidity Tiers

Liquidity Tiers group markets of similar risk into standardized risk parameters. Liquidity tiers specify the margin requirements needed for each market and should be determined based on the depth of the relative market's spot book as well as the token's market capitalization.

Current Liquidity Tiers include:

ID Name initial margin fraction maintenance fraction (what fraction MMF is of IMF) base position notional impact notional maintenance margin fraction (as is) impact notional (as is) Lower Cap (USDC Millions) Upper Cap (USDC Millions) 0 Large-Cap 0.05 0.6 1_000_000 USDC 500 USDC / IM 0.03 10_000 USDC None None 1 Mid-Cap 0.1 0.5 250_000 USDC 500 USDC / IM 0.05 5_000 USDC 15 25 2 Long-Tail 0.2 0.5 100_000 USDC 500 USDC / IM 0.1 2_500 USDC 2.5 5 3 Safety 1 .02 1_000 USDC 2500 USDC / IM 0.2 2_500 USDC 0.5 1 * Each market has aLower Cap * andUpper Cap * denominated in USDC. * Each market already has aBase IMF * . * At any point in time, for each market: *Define* Open Notional = Open Interest * Oracle Price * * * Scaling Factor = (Open Notional - Lower Cap) / (Upper Cap - Lower Cap) * * * IMF Increase = Scaling Factor * (1 - Base IMF) * * Then a market'sEffective IMF = Min(Base IMF + Max(IMF Increase, 0), 1.0) * The effective IMF is the base IMF while the Open Notional < Lower Cap, and increases linearly until Open Notional = Upper Cap, at which point the IMF stays at 1.0 (requiring 1:1 collateral for trading)

Governance has the ability to create and modify Liquidity Tiers as well as update existing markets' Liquidity Tier placements. (proto(opens in a new tab))

Updating a Live Market

This functionality allows the community to update parameters of a live market, which can be composed of 4 parts

- Updating a liquidity tier
- Perpetual (x/perpetuals
-), governance-updatable throughMsgUpdatePerpetualFeeParams
- (proto definition(opens in a new tab)
-)
- Market (x/prices
-), governance-updatable throughMsgUpdateMarketParam
- (proto(opens in a new tab)
-)
- Clob pair (x/clob
-), governance-updatable throughMsgUpdateClobPair
- (proto(opens in a new tab)
-)

Adding New Markets

The action of a governance proposal is defined by thelist of messages that are executed(opens in a new tab) when it's accepted. A proposal to add a new market should include the following messages (in this particular order):

MsgCreateOracle (create objects in x/prices) MsgCreatePerpetual (create object in x/perpetual) MsgCreatePerpetualClobPair (create object in x/clob) MsgDelayMessage (schedule a MsgSetClobPairStatus to enable trading in x/clob)

Safety

Spam Mitigation

To prevent spam on the orderbook and prevent the blockchain state from getting too large, governance has the ability to adjust:

- How many open orders a subaccount can have based on its equity tier.Proto(opens in a new tab)
- Order placement rate limits.Proto(opens in a new tab)

Bridge

Bridge Module

The Bridge Module is responsible for receiving bridged tokens from the Ethereum blockchain.

Governance has the ability to update:

- Event Parameters: Specifies which events to recognize and which tokens to mintProto(opens in a new tab)
- Proposal Parameters: Determines how long a validator should wait until it proposes a bridge event to other validators, and how many or often to propose new bridge events.Proto(opens in a new tab)
- Safety Parameters: Determines if bridging is enabled/disabled and how many blocks mints are delayed after being accepted by consensus.Proto(opens in a new tab)

Community Assets

Community Pool + Community Treasury

There are two addresses intended for managing funds owned by the community:

1. a Community Pool and
2. a Community Treasury.

The Community Pool is the recipient of any Community Tax that is implemented via the Distribution Module. The Community Pool is controllable by governance.

The Community Treasury is an account controlled by governance and can be funded via any account or module sending tokens to it.

CosmosSDK Default Modules:

For more information on default modules, head to theCosmos SDK official documentation(opens in a new tab). dYdX Chain inherits the same governance properties of any standard CosmosSDK modules that are present on dYdX Chain,