

# Adding cooldown from Position into Ponder Indexer

---

Ref. Smart Contracts: [Smart Contracts](#)

Ref. Github Repo: <https://github.com/Frankencoin-ZCHF/FrankenCoin>

A list of all the instances where the cooldown variable is assigned a new value in the Position.sol contract:

```
openPosition(...) method:

cooldown = block.timestamp + _duration;


_close() internal method:

cooldown = type(uint256).max;


_restrictMinting(uint256 period) internal method:

if (horizon > cooldown) {
    cooldown = horizon;
}
```

Indexer updated for:

- ☒ openPosition
- ☐ \_close
- ☐ \_restrictMinting

Who calls, from where does the calls come from?

## 1. openPosition

- function **MintingHub**:openPosition(...) public returns (address)
  - emits event **MintingHub:PositionOpened**( address indexed owner, address indexed position, address zCHF, address collateral, uint256 price );
- function **MintingHub**:clone(...) public validPos(position) returns (address)
  - emits event **MintingHub:PositionOpened**( address indexed owner, address indexed position, address zCHF, address collateral, uint256 price );

## 2. \_close

- function \_considerClose(uint256 collateralBalance) internal
  - function notifyChallengeAverted(uint256 size) external onlyHub
    - function **MintingHub**:avertChallenge(Challenge memory \_challenge, uint32 number, uint256 liqPrice, uint256 size) internal
      - function **MintingHub**:bid(uint32 \_challengeNumber, uint256 size, bool postponeCollateralReturn) external
        - emits event **MintingHub:ChallengeAverted**(address indexed position, uint256 number, uint256 size);
  - function \_withdrawCollateral(address target, uint256 amount) internal returns (uint256)
    - emits event MintingUpdate(uint256 collateral, uint256 price, uint256 minted, uint256 limit);
- function deny(address[] calldata helpers, string calldata message) external
  - emits event PositionDenied(address indexed sender, string message); // emitted if closed by governance

### 3. \_restrictMinting

- function adjustPrice(uint256 newPrice) public onlyOwner noChallenge
  - event MintingUpdate(uint256 collateral, uint256 price, uint256 minted, uint256 limit);
- function notifyChallengeAverted(uint256 size) external onlyHub (**<<< DUPLICATED >>>**)
  - function **MintingHub**:avertChallenge(Challenge memory \_challenge, uint32 number, uint256 liqPrice, uint256 size) internal
    - function **MintingHub**:bid(uint32 \_challengeNumber, uint256 size, bool postponeCollateralReturn) external
      - emits event **MintingHub:ChallengeAverted**(address indexed position, uint256 number, uint256 size);
- function notifyChallengeSucceeded(address \_bidder, uint256 \_size) external onlyHub returns (address, uint256, uint256, uint32)
  - function **MintingHub**:\_finishChallenge (...)
    - emits event **MintingHub:ChallengeSucceeded**( address indexed position, uint256 number, uint256 bid, uint256 acquiredCollateral, uint256 challengeSize );

### Duplicated code section

```
function notifyChallengeAverted(uint256 size) external onlyHub {
    challengedAmount -= size;
```

```
    _restrictMinting(1 days);  
    _considerClose(_collateralBalance());  
}
```

## List of emitted Events to catch a change in cooldown

1. ☒ openPosition (create)
  - ☒ MintingHub:PositionOpened (from openPosition)
  - ☒ MintingHub:PositionOpened (from clonePosition)
2. ☐ \_close (update)
  - ☐ MintingHub:ChallengeAverted
  - ☐ Position:MintingUpdate
  - ☐ Position:PositionDenied
3. ☐ \_restrictMinting (update)
  - ☐ Position:MintingUpdate
  - ☐ MintingHub:ChallengeAverted
  - ☐ MintingHub:ChallengeSucceeded