

XFI-STAKING DOCUMENTATION

Lumos Labs Team

October 2023

1 Definitions

LP	LP-XFI/TKN(any token paired with XFI)
ST	esXFI
BP	Bonus points
TS_x	Total supply of x
RR_x	Reward rate of x

2 Introduction

In the initial phase, users are only permitted to deposit LP-XFI/TKN tokens, commonly known as LP tokens, into the smart contract. These deposits serve as the foundation for generating returns. The distribution of rewards, encompassing both esXFI and wETH, is intricately linked to the number of esXFI tokens held by the user and their Bonus Points. Consequently, a user's allocation of rewards is influenced by their accumulated esXFI tokens, Bonus Points and LP.

2.1 esXFI

Users receive XFI, which we will refer to as NC (native currency), as incentive rewards wrapped within esXFI, hereinafter denoted as ST (staking token). These ST rewards are generated through a Proof of Stake (PoS) algorithm, where the Foundation stakes MPX, and all the earned NC is distributed as rewards. Once the rewards are credited, users have the flexibility to choose between two options:

- Stake their ST as a regular LP, thereby enhancing their portion of the generated yield.
- Commence the ST vesting period in order to access and withdraw NC.

2.2 Bonus Points

Bonus Points, which we will refer to as BP, should be distinguished as they are not rewards or tokens but rather a mechanism designed to incentivize stable XFI stakers.

BP is earned by stakers at an annual rate of 100% APR based on their LP stake size. For instance, if a user stakes 100 LP for a year, they will accrue 100 BP.

It is important to note that if a user decides to unstake their LP, their accumulated BP will reset.

2.3 Rewards distribution

The rewards that a user receives are directly proportional to the quantity of LP they have staked, the count of BP they possess, and their ST holdings. It's worth noting that ST has an auto-compounding feature, enhancing its growth over time. BP, on the other hand, compounds for the user every time he switch pools.

The allocation of rewards among all users is determined by the following formula:

$$\frac{LP + BP + ST}{TS_LP + TS_BP + TS_ST} = reward$$

From the information provided, several key conclusions can be drawn:

First, staking LP for an extended period is in a user's best interest as it increases their share of rewards over time. Second, users should exercise caution when contemplating LP unstaking since it results in the loss of all accrued Bonus Points (BP), which significantly reduces their rewards. Lastly, it's important to note that BP cannot be transferred and are uniquely associated with the staking user.

2.4 Main idea

As we observe, the proportions of users in the distribution of rewards remain constant even when there are changes in the pool configuration. Furthermore, it's essential to recognize that over a given time period t the total quantity of all tokens (with BP being taken into account in this total) represented by $TS_BP + TS_LP + TS_ST$ will evolve as follows: $TS_BP + TS_LP + TS_ST + RR_n * t$.

To better comprehend how a user's earnings fluctuate at moments of pool adjustments, typically when any user invokes a mutative function to alter the token distribution proportions, we can trace how the total number of their tokens increases m times, where

$$m = \frac{TS_BP + TS_LP + TS_ST + RR_n * t}{TS_BP + TS_LP + TS_ST}$$

$$t = block.timestamp - lastUpdateTime$$

3 Earning ST

While the proportions between pool changes remain stable, it remains uncertain how a user's share will be affected when another user modifies the pool. To ascertain this, we must multiply the total number of a user's tokens by m after each pool alteration. To achieve this in a gas-efficient manner, we introduce two variables: *nativeMultiplierStored*, which signifies the product of all m values (this variable updates following each *updateReward* action), and *userNativeMultiplierPaid* for each user, representing the product of all m values preceding the user's action. Consequently, the formula for calculating rewards in ST appears as follows:

$$balanceST = \frac{(LP + ST + BP) \times nativeMultiplierStored}{userNativeMultiplierPaid} - LP - BP$$

4 Earning wETH

The key distinction from the prior version is the exclusion of wETH's contribution to the reward distribution share. As a result, a straightforward multiplication of all values together will yield an incorrect result. Instead, we must aggregate the earned amounts into a unified value. Thus, to provide a succinct and formula-centric explanation:

$$tokenEarned = RR_t \times bal[acc] \times \sum_{i=1}^n \left(\frac{t_i \times \prod_{k=1}^i (m_k)}{TS_i} \right)$$

$$bal[acc] \rightarrow balanceLP[account] + balanceBP[account] + balanceST[account]$$

$$RR_t \rightarrow tokenRewardRate$$

$$\begin{aligned}
t_i &\rightarrow \text{block.timestamp} - \text{lastUpdateTime}_i \\
TS_i &\rightarrow TS_LP + TS_BP + TS_ST \text{ (in lastUpdateTime}_i \text{ moment)} \\
m_k &\rightarrow \text{how many times bal[acc] grow over time } t_i
\end{aligned}$$

The development of this formula stems from a human-driven approach. It's crucial to grasp that the distribution ratios remain unchanged between pool modifications. Consequently, the need to adjust a user's earned rewards arises mainly when there is a shift in the pool configuration. In this context, we can interpret the user's earnings as follows:

$$tokenEarned = \frac{bal[acc]}{TS} \times RR_t \times t_1 + \frac{bal'[acc]}{TS'} \times RR_t \times t_2 + \dots$$

$$bal'[acc] \rightarrow \text{user's balance after } t_1$$

$$TS' \rightarrow \text{total supply after } t_1$$

To calculate this efficiently in terms of gas usage, we can introduce two essential variables: *tokenMultiplierStored*, which maintains and updates the cumulative sum derived from the formula, and *balanceMultiplierStored*, which retains and updates the product obtained from the formula.

4.1 Vesting

XFI vesting has the following conditions:

1. Vesting occurs over a one-year period with a gradual unlock of tokens (similar to Aragon DAO vesting).
2. To start ST vesting, a user must stake a certain amount of LP (10 LP per 1 ST).
3. User gradually get XFI from unlocked esXFI.
4. If a user unstakes a certain amount of LP tokens, they will receive the following amount of NC every second:

$$NC = \min(VST, \frac{LP}{10}) * \frac{t}{ONE_YEAR_IN_SECS}$$

$$t \rightarrow \text{time interval between pool updates}$$

$$VST \rightarrow \text{user balance in vested ST}$$