TxPipe Docs

Cardano-specific Features

This guide covers blockchain-specific features in Tx3 for the Cardano blockchain.

Protocol Parameters

```
// Access protocol parameters
pparams.min_fee_coefficient
pparams.min_fee_constant
pparams.coins_per_utxo_byte
```

Native Scripts

```
// Define native script
policy TimeLock = import("validators/vesting.ak");

// Use native script
tx lock(until: Int) {
    input source {
        from: TimeLock,
        min_amount: Ada(amount),
    }
}
```

Certificates

```
cardano {
    certificates: [
        StakeRegistration { ... },
        StakeDelegation { ... },
        StakeDeregistration { ... },
    ]
}
```

Withdrawals

```
cardano {
    withdrawals: [
        (StakeCredential, Int),  // (stake credential, amount)
    ]
}
```

Collateral

```
cardano {
    collateral: input {
        from: User,
        min_amount: Ada(collateral_amount),
    }
}
```

Common Patterns

Stake Registration

```
tx register_stake(
    stake_credential: StakeCredential
) {
    input source {
        from: User,
        min_amount: Ada(registration_fee),
    }

    cardano {
        certificates: [
            StakeRegistration {
                credential: stake_credential,
            }
        ]
    }
}
```

Stake Delegation

```
tx delegate_stake(
    stake_credential: StakeCredential,
    pool_id: PoolId
) {
    input source {
        from: User,
        min_amount: Ada(delegation_fee),
    }

    cardano {
        certificates: [
            StakeDelegation {
                credential: stake_credential,
                pool_id: pool_id,
            }
        ]
    }
}
```

Reward Withdrawal

```
tx withdraw_rewards(
    stake_credential: StakeCredential,
    amount: Int
) {
    input source {
        from: User,
        min_amount: Ada(withdrawal_fee),
    }

    cardano {
        withdrawals: [
            (stake_credential, amount)
        ]
    }
}
```

Common Use Cases

Stake Pool Registration

```
tx register_pool(
    pool_params: PoolParams
) {
    input source {
        from: Operator,
        min_amount: Ada(registration_fee),
    }

    cardano {
        certificates: [
            PoolRegistration {
                params: pool_params,
            }
        ]
    }
}
```

Stake Pool Retirement

```
tx retire_pool(
    pool_id: PoolId,
    epoch: Int
) {
    input source {
        from: Operator,
        min_amount: Ada(retirement_fee),
    }

    cardano {
        certificates: [
            PoolRetirement {
                pool_id: pool_id,
                epoch: epoch,
            }
        ]
    }
}
```

Multi-Certificate Transaction

```
tx multi_cert(
    stake_cred: StakeCredential,
    pool_id: PoolId
) {
    input source {
        from: User,
        min_amount: Ada(total_fee),
    }

    cardano {
        certificates: [
            StakeRegistration {
                credential: stake_cred,
            },
            StakeDelegation {
                credential: stake_cred,
                pool_id: pool_id,
            }
        ]
    }
}
```

Network-Specific Features

Testnet Support

```
// Network selection
network = "testnet"

// Testnet-specific parameters
pparams.testnet = true
```

Mainnet Support

```
// Network selection
network = "mainnet"

// Mainnet-specific parameters
pparams.mainnet = true
```