

# EOS开发入门体验

前面一篇《EOS简介与环境搭建》对EOS作了简单介绍和环境搭建，并没有涉及到编译和部署智能合约。EOS的智能合约是基于C++语言的，感觉门槛稍微有点高吧。不过，本篇还没有涉及到EOS智能合约的编写。

本篇的主要目的是记录下学习的过程，避免大家少走弯路少踩坑。还有好多不懂的地方，可能会有些描述不当的地方，还请大家多指正。话不多说，开始带领大家来一场EOS的开发入门体验。

## EOS版本说明

EOS的版本特别多，可谓五花八门，可能会遇到各种坑。

## EOS官方版本列表

EOSIO / eos

Watch

1,206

Star

7,542

Fork

1,901

<> Code

Issues 609

Pull requests 13

Projects 0

Wiki

Insights

An open source smart contract platform <https://eosio.github.io/eos/>

7,442 commits

78 branches

51 releases

123 contributors

MIT

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

Switch branches/tags

Find a tag...

BranchesTags

v1.0.0

help

dawn-v4.2.0

dawn-v4.1.0

dawn-v4.0.0

dawn-v3.0.0

dawn-v2.0.0

dawn-v1.0.0

SuperDawn-2018-03-18

SuperDawn-2018-03-04

SuperDawn-2018-03-02

STAT-2017-10-04

Latest commit a9526dc 8 hours ago

base pipeline timeouts 2 days ago

te other version numbers to 1.0 and cleanup 14 hours ago

er README updates [skip-ci] 11 hours ago

sue #3740 12 hours ago

me eosiod to nodeos and eosioc to cleos and update docs and scripts. 2 months ago

te to 4.1 14 days ago

ert binaryen asserts into throw #3498 4 days ago

ate wiki images and remove unused Doxyfile. 13 hours ago

e pull request #3749 from EOSIO/gh#3728-eosiocpp-crash 10 hours ago

y\_plugin: fixing potential integer overflow #3735 11 hours ago

e branch 'master' into ChangeVersion 14 hours ago

ged to one line in main build script 2 days ago

e pull request #3742 from EOSIO/repository-root-cleanup 10 hours ago

-output-on-failure to code coverage 25 days ago

bios-boot-tutorial: 200,000 accounts 13 hours ago

就在今天（6月2日）早上，EOSIO发布了1.0版本，对Dawn 4.2做了一些小改动。该版本加入了一个可选的P2P网络协议，能够支持多线程网络传输以及更快的同步。

就拿EOS的博客来说，网上的博客有些是直接从官方文档翻译过来的，自己可能都没有试验过。

有些博客上文不接下文，命令格式都有问题，或者中途跳过了某些操作，可能到了哪一步都执行不了。

比如很多博客讲EOS提供的currency智能合约部署，但是这个currency智能合约在2018年5月11号发布的DAWN 4.0版本就移除了。于是，如果Git下载的是最新的EOS源码，部署currency合约的时候就会出错。当然，也可以下载指定的版本来编译学习。

本文是在最新版本上实验的，这个也是遇到了部署currency合约的坑才发现的。可以跳转"2、部署BIOS合约"的一节查看，发现里面的"contracts"文件夹没有"currency"合约。

## 钱包的使用

### 1、创建钱包

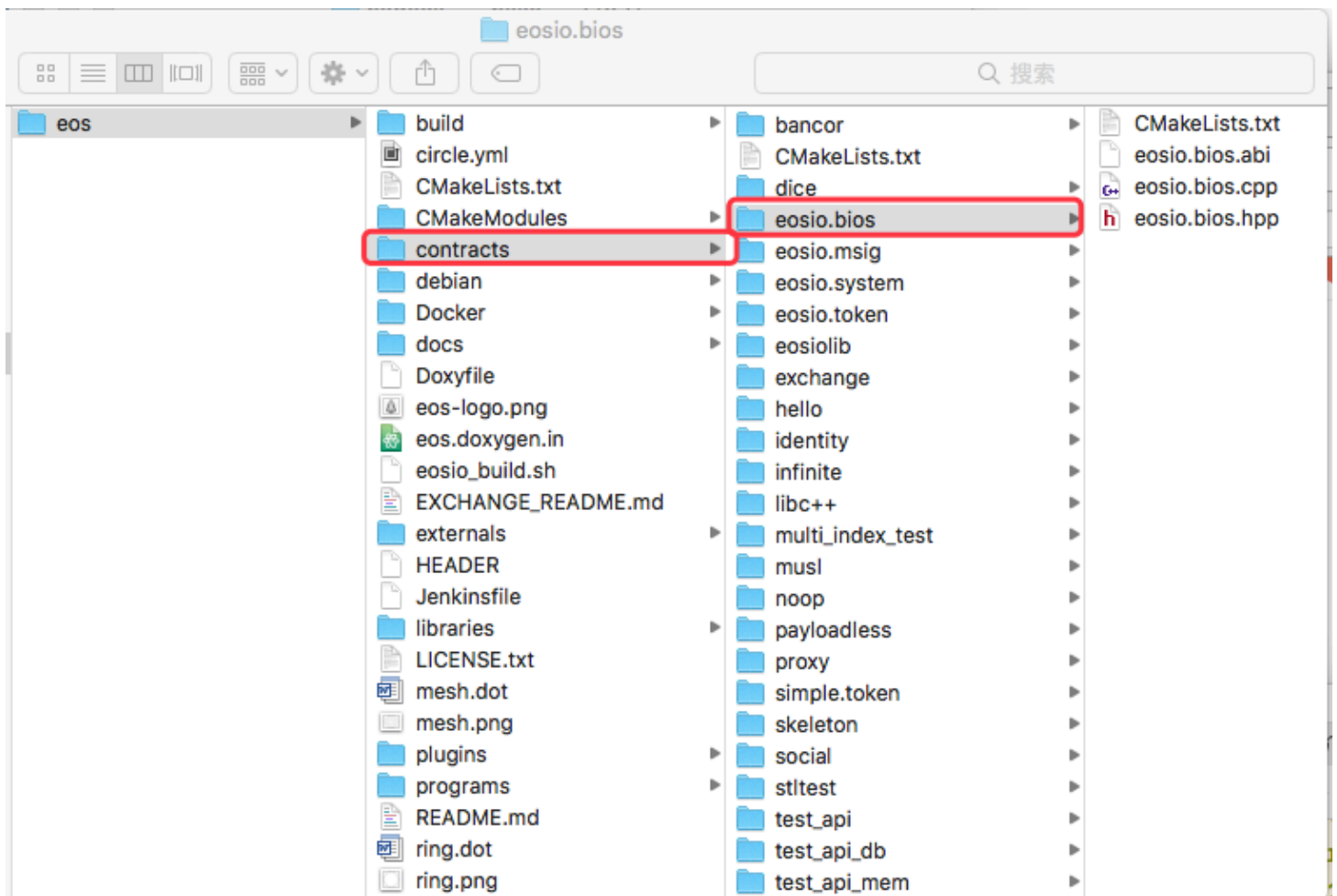
进入到"cleos"组件的根目录，然后执行"./cleos wallet create"命令

```
wenzildeiMac:cleos wenzil$ ./cleos wallet create  
Creating wallet: default  
Save password to use in the future to unlock this wallet.  
Without password imported keys will not be retrievable.  
"PW5JiXgmkmpT62QEv8aqX56pG8WT2daXnDM21wzmMKfdHQfbT2eKw"
```

"PW5JiXgmkmpT62QEv8aqX56pG8WT2daXnDM21wzmMKfdHQfbT2eKw"是随机生成的钱包私钥，需要记录下来并保存好。

### 2、部署Bios合约

EOS自带了很多智能合约，目录位于"eos/contracts"目录下，eosio.bios是其中一个。



确保EOS的区块链节点组件nodeos在运行状态，否则会提示如下内容：

```
wenzildeiMac:cleos wenzil$ ./cleos set contract eosio ../../contracts/eosio.bios -p eosio
Reading WAST/WASM from ../../contracts/eosio.bios/eosio.bios.wasm...
Using already assembled WASM...
Publishing contract...
Failed to connect to nodeos at http://localhost:8888/; is nodeos running?
```

成功的运行结果：

```
wenzildeiMac:cleos wenzil$ ./cleos set contract eosio ../../contracts/eosio.bios -p eosio
Reading WAST/WASM from ../../contracts/eosio.bios/eosio.bios.wasm...
Using already assembled WASM...
Publishing contract...
executed transaction: b8f2f986a3bd6b2b7d237a347ce92117fabfa0adb713f38437c4cea3a543bb33 3712 bytes 7143 us
# eosio <= eosio::setcode {"account":"eosio","vmtype":0,"vmversion":0,"code":"0061736d010000001621260037f7e7f0060057f7e7e7e...
# eosio <= eosio::setabi {"account":"eosio","abi":"0e656f73696f3a3a6162692f312e30050c6163636f756e745f6e616d65046e616d650f7065...
```

**warning: transaction** executed locally, but may **not** be confirmed **by** the network yet

### 3、创建一组密钥

```
wenzildeiMac:cleos wenzil$ ./cleos create key  
Private key: 5K55ERTgLcR9RK5K7fagwJaHXCvnZJy5cMHh6xmNgLUXvneTTzr  
Public key: EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51zzat
```

### 4、导入私钥到钱包

命令格式为：

```
./cleos wallet import [private_key]
```

private\_key: 为上一步生成的私钥

操作如下：

```
wenzildeiMac:cleos wenzil$ ./cleos wallet import 5K55ERTgLcR9RK5K7fagwJaHXCvnZJy5cMHh6xmNgLUXvneTTzr  
imported private key for: EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51zzat
```

## 账户的使用

### 5、创建用户账户

使用导入私钥到钱包生成的密钥来创建用户账户，以下演示创建两个账户user1和user2

命令格式为：

```
./cleos create account eosio [account_name] [owner_key] [active_key]
```

cleos create 命令需要两个密钥，为了方便，这里给它两个相同的密钥，都是上面生成的私钥。

account\_name: 为要创建的账户名

owner\_key: 作用是掌握该账户所有控制权限

active\_key: 作用是只掌握该账户资金访问权限

操作如下：

```
wenzildeiMac:cleos wenzil$ ./cleos create account eosio user1 EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51zzat EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51zzat
executed transaction: c0b7ff03972c5829a6fb3b8a3d422f5f08b8213c95c1883fd5a641e5c55c029b 200 bytes 250 us
# eosio <= eosio::newaccount {"creator":"eosio","name":"user1","owner":{"threshold":1,"keys":[{"key":"EOS6Ps9FM3A6yi5JgVZ7hPzQU4g...
warning: transaction executed locally, but may not be confirmed by the network yet

wenzildeiMac:cleos wenzil$ ./cleos create account eosio user2 EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51zzat EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51zzat
executed transaction: 067e145a00df9ad9b8cc5385b06d83251fa535311470ee512a66e0bf0fc41578 200 bytes 233 us
# eosio <= eosio::newaccount {"creator":"eosio","name":"user2","owner":{"threshold":1,"keys":[{"key":"EOS6Ps9FM3A6yi5JgVZ7hPzQU4g...
warning: transaction executed locally, but may not be confirmed by the network yet
```

## 6、查看对应的账户信息

命令格式为：

```
./cleos get account [account_name]
```

操作如下：

```
wenzildeiMac:cleos wenzil$ ./cleos get account user1
privileged: false
permissions:
  owner      1:      1 EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51zzat
  active     1:      1 EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51zzat
memory:
  quota:      -1 bytes  used:      2.66 Kb

net bandwidth: (averaged over 3 days)
  used:      -1 bytes
  available: -1 bytes
  limit:     -1 bytes

cpu bandwidth: (averaged over 3 days)
  used:      -1 us
  available: -1 us
```

limit: -1 us

## 部署和调用智能合约

### 7、部署Token合约

命令格式为：

```
./cleos set contract [account_name] ../../contracts/eosio.token -p [account_name]
```

[account\_name]是要部署的账户，这里替换为"user1"。

```
wenzildeiMac:cleos wenzil$ ./cleos set contract user1 ../../contracts/eosio.token -p user1
Reading WAST/WASM from ../../contracts/eosio.token/eosio.token.wasm...
Using already assembled WASM...
Publishing contract...
executed transaction: 415e0dd1776d5af0edc82609c5614252f8ab0a62df1214934007b9d4b6ce8919 8104 bytes 16915 us
# eosio <= eosio::setcode {"account":"user1","vmtype":0,"vmversion":0,"code":"0061736d01000000017e1560037f7e7f0060057f7e7e7f7f...
# eosio <= eosio::setabi {"account":"user1","abi":"0e656f73696f3a3a6162692f312e30010c6163636f756e745f6e616d65046e616d65050874...
warning: transaction executed locally, but may not be confirmed by the network yet
```

### 8、调用Token合约的create函数

命令为：

```
./cleos push action user1 create '{"issuer":"eosio", "maximum_supply":"1000000000.0000 EOS", "can_freeze":0, "can_recall":0, "can_whitelist":0}' -p user1
```

创建一个名称为"EOS"的Token（代币），总发行量为10亿。

```
wenzildeiMac:cleos wenzil$ ./cleos push action user1 create '{"issuer":"eosio", "maximum_supply":"1000000000.0000 EOS", "can_freeze":0, "can_recall":0, "can_whitelist":0}' -p user1
executed transaction: feef0adec95643f059085f813491b884d2058d7c987e02678ef359524e887e6c 120 bytes 378 us
```

```
#          user1 <= user1::create          {"issuer":"eosio","maximum_s
upply":"1000000000.0000 EOS"}
warning: transaction executed locally, but may not be confirmed by the netwo
rk yet
```

## 9、发行代币

命令为：

```
./cleos push action user1 issue '["user1", "100.0000 EOS","memo"]' -p eosio
```

给用户user1发100个EOS币，留言是"memo"

```
wenzildeiMac:cleos wenzil$ ./cleos push action user1 issue '["user1", "100.
0000 EOS","memo"]' -p eosio
executed transaction: a1dcff4887555bd037ffad0d2877193ed2f56195b07c5ac220fd13
c5846e54d6 128 bytes 33617 us
#          user1 <= user1::issue          {"to":"user1","quantity":"10
0.0000 EOS","memo":"memo"}
#          user1 <= user1::transfer          {"from":"eosio","to":"user1"
,"quantity":"100.0000 EOS","memo":"memo"}
#          eosio <= user1::transfer          {"from":"eosio","to":"user1"
,"quantity":"100.0000 EOS","memo":"memo"}
warning: transaction executed locally, but may not be confirmed by the netwo
rk yet
```

现在user1有100个EOS币了。

10、如果出现了如下情况，需要对钱包进行解锁。

```
Error 3120003: Locked wallet
Ensure that your wallet is unlocked before using it!
Error Details:
You don't have any unlocked wallet!
```

执行如下命令，然后输入钱包的私钥，也就是创建钱包时返回的私钥。

```
wenzildeiMac:cleos wenzil$ ./cleos wallet unlock
password:
```

## 11、转账交易

命令为：

```
./cleos push action user1 transfer '["user1", "user2", "20.0000 EOS", "哥送你20个EOS币"]' -p user1
```

通过user1向user2转20个ESO币

```
wenzildeiMac:cleos wenzil$ ./cleos push action user1 transfer '["user1", "user2", "20.0000 EOS", "哥送你20个EOS币"]' -p user1
executed transaction: 4ab89c1618eb90c8a3fd56d6182f13510b9e5ac50f09fe8a61b96ae88346bf51 152 bytes 677 us
#          user1 <= user1::transfer {"from":"user1","to":"user2",
,"quantity":"20.0000 EOS","memo":"哥送你20个EOS币"}
#          user2 <= user1::transfer {"from":"user1","to":"user2",
,"quantity":"20.0000 EOS","memo":"哥送你20个EOS币"}
warning: transaction executed locally, but may not be confirmed by the network yet
```

## 12、查看账户信息

```
wenzildeiMac:cleos wenzil$ ./cleos get table user1 user1 accounts
{
  "rows": [{
    "balance": "80.0000 EOS"
  }],
  "more": false
}

wenzildeiMac:cleos wenzil$ ./cleos get table user1 user2 accounts
{
  "rows": [{
    "balance": "20.0000 EOS"
  }],
  "more": false
}
```

现在user1有80个EOS，user2有20个EOS代币。