目标

实现一个受管控的专用公链，支持跨链、快速转账、资产token化、托管、结算等业务。
选取EOS公链代码作为扩展的基础，主要因为EOS公链具有以下特性，
1，账号授权机制相对完善，
2，打包速度相对快速，
3，公链能耗相对比较低，
通过简单的扩展，即可支持业务需求。

编译

```
git clone https://github.com/EOSIO/eos --recursive
git checkout -b v2.0.5

git clone --recursive https://github.com/eosio/eosio.cdt
git checkout -b v1.7.0-rc1

git clone --recursive https://github.com/EOSIO/eosio.contracts.git
git checkout -b v1.9.1

git clone https://github.com/myshare2020/uac --recursive
```

下载对应版本的eos代码，使用uac代码覆盖相应的代码。

主要扩展功能

1，默认禁止账号设置合约，约束发布应用的权限。
2，增加应用管理员角色，联合管理应用、代币等资源。
3，增加应用属性，约束应用的使用。
4，增加代币属性，约束代币的使用。

扩展内建账号

eosio.app 应用管理员

扩展接口命令

1.设置账号属性：
需要应用管理员授权。
bindattrs <account>
account 用户账号
--can_set_contract 允许账号设置合约

2.设置应用属性：
需要操作账号、应用账号、应用管理员授权。
bindapp <account> <opr_code> <appid>
account 操作账号
opr_code 操作，modify修改属性，transfer转移应用所有权，create创建应用
appid 应用id
--has_memo 修改memo
--memo 应用描述(800字节)
--has_expire 修改expire
--expire 应用过期时间（ISO时间格式）

创建应用
bindapp apponeowner1 create 1000 --has_memo --memo "test application" --has_expire --expire "2030-01-01T00:00:00" -p eosio.app -p apponeowner1

转移应用所有权
bindapp apponeowner2 transfer 1000 --has_memo --memo "new application" --has_expire --expire "2020-01-01T00:00:00" -p eosio.app -p apponeowner1 -p apponeowner2

应用过期后，不再需要应用所有者授权，相当于应用所有者放弃了所有权，应用所有者需提前续期。

3.设置代币属性：
需要操作账号、应用账号、应用管理员授权。
bindsym <account> <opr_code> <sym>
account 操作账号
opr_code 操作，modify修改属性，transfer转移代币所有权
sym 代币符号

--has_tollor 修改tollor
--tollor 收费账号
--has_issuer 修改issuer
--issuer 发行账号
--has_fee 修改收费规则
--minval 开始收费转账数额
--minfee 最小费
--level 阶梯跨度
--delta 阶梯费差
--maxfee 最大费

如果设置了收费规则，在代币转账时收取UAC作为转账费，
转账数额value>=minval开始收费，
  固定费，level==0，minfee
  阶梯费，level>0，(value-minval)/level*delta+minfee，最大maxfee

4.查询代币属性
get sym <sym>
sym 代币符号
返回sym,appid,tollor,issuer,minval,minfee,level,delta,maxfee

转移代币所有权
// 4,GBTC
bindsym apponeowner3 transfer GBTC --has_tollor --tollor apponetollr3 --has_issuer --issuer apponeissue3 --has_fee --minval 1000000 --minfee 10 -p eosio.app -p apponeowner2 -p apponeowner3

5.查询应用属性
get app <appid>
appid 应用id
返回appid,owner,created,expire,memo

6.查询收费数额
get fee <sym> <value>
sym 代币符号
value 代币转账数额
返回应收UAC数额

扩展合约指令

1.获取公链本币symbol
uint64_t get_core_sym()
返回本币符号和精度

2.获取收费额
name get_app_fee(uint64_t sym, int64_t value, int64_t* fee)
sym 代币符号
value 转账数额
fee 应收费数额（输出）
返回收费账号

3.关联代币的应用账号和发行账号
void new_app_sym(uint64_t sym, uint64_t owner, uint64_t issuer)
sym 代币符号
owner 应用账号
issuer 发行账号

4.获取代币发行者
name get_sym_issuer(uint64_t sym)
sym 代币符号
返回发行账号

5.创建代币
void create2(name owner, name issuer, asset maximum_supply)
owner 应用账号
issuer 发行账号
maximum_supply 最大发行量

cleos push action eosio.token create2 '["apponeowner1","apponeissue1","10000000000.0000 GBTC"]' -p eosio.app -p apponeowner1