此文档用来说明智能合约首次部署以及更新的步骤。

# 首次部署

## 购买内存

cleos --wallet-url http://127.0.0.1:6666 system buyram useraaaaaaag useraaaaaaaj "10 SYS"

## 部署合约

Bac游戏合约部署：

cleos --wallet-url http://127.0.0.1:6666 set contract gamemallards ./ mallard.wasm mallard.abi

Sic游戏合约部署：

cleos --wallet-url http://127.0.0.1:6666 set contract game12lizard ./ lizard.wasm lizard.abi

## 发币（可选）

cleos --wallet-url http://127.0.0.1:6666 set contract useraaaaaaaj /home/eos/eosio.contract/build/eosio.token/ eosio.token.wasm eosio.token.abi

cleos --wallet-url http://127.0.0.1:6666 push action useraaaaaaaj create '["useraaaaaaai","100000000.0000 TES"]' -p useraaaaaaaj

cleos --wallet-url http://127.0.0.1:6666 push action useraaaaaaaj issue '["useraaaaaaab","100000.0000 TES","xxx"]' -p useraaaaaaai

cleos --wallet-url http://127.0.0.1:6666 push action useraaaaaaaj issue '["useraaaaaaac","10000.0000 TES","xxx"]' -p useraaaaaaai

cleos --wallet-url http://127.0.0.1:6666 push action useraaaaaaaj issue '["useraaaaaaad","10000.0000 TES","xxx"]' -p useraaaaaaai

cleos --wallet-url http://127.0.0.1:6666 get currency balance useraaaaaaaj useraaaaaaab "TES"

cleos --wallet-url http://127.0.0.1:6666 get currency balance useraaaaaaaj useraaaaaaac "TES"

cleos --wallet-url http://127.0.0.1:6666 get currency balance useraaaaaaaj useraaaaaaad "TES"

## 初始化添加合约支持币种

cleos --wallet-url http://127.0.0.1:6666 push action gamemallards initsymbol '["useraaaaaaaj","TES","0.1000 TES"]' -p gamemallards

cleos --wallet-url http://127.0.0.1:6666 push action gamemallards initsymbol '["eosio.token","SYS","0.1000 SYS"]' -p gamemallards

# 更新合约

## 备份原合约

为防止误操作，部署了新结构合约无法恢复，更新前要备份原合约的wasm、abi。

## 调用更新接口

SC::upgrading(1)

修改一个标志位upgrading为true，默认false。（当upgrading为true时，非托管下dealerseed以及托管下的serverseed两个合约均不可被执行，也就是说，当标志位upgrading为true时，等待所有桌本轮结束，均不可开启新一轮）

## 校验tablesinfo

校验所有table的status是否只有两种状态：ROUND\_END或PAUSED

## 导出tablesinfo表到本地

Server端mysql储存当前所有链上状态表数据，通过getTableRows。

## 删除tablesinfo

Tablesinfo数据全部导出成功后，调用clear12cache接口删除所有链上状态表数据。

## 部署新结构合约

部署包含数据结构修改的新合约

## 导入数据到新结构

调用import12data，按照参数列表传入所有桌相关字段（table fields），轮次字段数据不需要，因为上一轮已经结束了，新的一轮未开始。

**注意**：导入时，每桌字段upgrading标志位写死的仍旧是true，以保证全部数据正常以后，手动控制结束更新。

## 更新完成

SC::upgrading(0)，将所有桌的upgrading标志位置为false，更新完成，可以正常运行了。