比特币源码研读系列3 创建tx(2)

上篇我们讲到了创建tx需要做下面这些事，并且对步骤1 Find my spendable money中的一些重要函数进行了解读。

1. **Find my spendable money**
2. **创建tx。**
3. **广播tx**

接着上文说，我们继续来看步骤1的主体方法。

1. AvailableCoins方法是在钱包中找到所有可用的coins。它是步骤1的主要方法，实现的是find my spendable money。代码有点长，我们慢慢看。入参vCoins保存着找寻到的cons。

*void CWallet::AvailableCoins(std::vector<COutput> &vCoins, bool fOnlySafe, const CCoinControl \*coinControl, const CAmount &nMinimumAmount, const CAmount &nMaximumAmount, const CAmount &nMinimumSumAmount, const uint64\_t &nMaximumCount, const int &nMinDepth, const int &nMaxDepth) const*

*{*

*vCoins.clear();*

*{*

*LOCK2(cs\_main, cs\_wallet);*

*CAmount nTotal = 0;*

*for (std::map<uint256, CWalletTx>::const\_iterator it = mapWallet.begin(); it != mapWallet.end(); ++it)****//遍历钱包中的tx***

*{*

*const uint256& wtxid = it->first;*

*const CWalletTx\* pcoin = &(\*it).second;*

*if (!CheckFinalTx(\*pcoin))****//* *CheckFinalTx返回false表示tx还处于锁定状态，不可用。***

*continue;*

*if (pcoin->IsCoinBase() && pcoin->GetBlocksToMaturity() > 0)****//如果tx是coinbase，则需要检查tx的成熟度。Coinbase是tx的一种，没有inputs只有outputs，一般是系统给予的奖励，比如挖矿奖励的50btc就是在coinbase中。目前要求的是coinbase所在Block在主链的深度nDepth大于100时，该coin才可用。***

*continue;*

*int nDepth = pcoin->GetDepthInMainChain();****//tx在主链中的深度***

*if (nDepth < 0)*

*continue;*

*// We should not consider coins which aren't at least in our mempool*

*// It's possible for these to be conflicted via ancestors which we may never be able to detect*

*if (nDepth == 0 && !pcoin->InMempool())****//不在Block且不在mempool中则不可用。***

*continue;*

*bool safeTx = pcoin->IsTrusted();****//上一篇我们介绍过IsTrust方法，当true时表示该coin可spend。***

***//如果tx有做过标记，替换别的tx或者被别的tx替换过，且还没在Block中则不可用。***

*if (nDepth == 0 && pcoin->mapValue.count("replaces\_txid")) {*

*safeTx = false;*

*}*

*if (nDepth == 0 && pcoin->mapValue.count("replaced\_by\_txid")) {*

*safeTx = false;*

*}*

*if (fOnlySafe && !safeTx) {*

*continue;*

*}*

*if (nDepth < nMinDepth || nDepth > nMaxDepth)*

*continue;*

*for (unsigned int i = 0; i < pcoin->tx->vout.size(); i++) {****//检查tx的所有outputs，看看里面还有多少coin可用（比如找的零钱，以及帐号间互转）。***

*if (pcoin->tx->vout[i].nValue < nMinimumAmount || pcoin->tx->vout[i].nValue > nMaximumAmount)*

*continue;*

*if (coinControl && coinControl->HasSelected() && !coinControl->fAllowOtherInputs && !coinControl->IsSelected(COutPoint((\*it).first, i)))*

*continue;*

*if (IsLockedCoin((\*it).first, i))****//coin是否是锁定的***

*continue;*

*if (IsSpent(wtxid, i))****//是否已经spent***

*continue;*

***//上一篇方法1.需要对tx执行script来解码，确认是不是属于我的。如果is mine且类型是下面几种则可用，把tx放入到vcoins数组中。***

*isminetype mine = IsMine(pcoin->tx->vout[i]);*

*if (mine == ISMINE\_NO) {*

*continue;*

*}*

*bool fSpendableIn = ((mine & ISMINE\_SPENDABLE) != ISMINE\_NO) || (coinControl && coinControl->fAllowWatchOnly && (mine & ISMINE\_WATCH\_SOLVABLE) != ISMINE\_NO);*

*bool fSolvableIn = (mine & (ISMINE\_SPENDABLE | ISMINE\_WATCH\_SOLVABLE)) != ISMINE\_NO;*

*vCoins.push\_back(COutput(pcoin, i, nDepth, fSpendableIn, fSolvableIn, safeTx));*

*// Checks the sum amount of all UTXO's.*

*if (nMinimumSumAmount != MAX\_MONEY) {*

*nTotal += pcoin->tx->vout[i].nValue;*

*if (nTotal >= nMinimumSumAmount) {*

*return;*

*}*

*}*

*// Checks the maximum number of UTXO's.*

*if (nMaximumCount > 0 && vCoins.size() >= nMaximumCount) {*

*return;*

*}*

*}*

*}*

*}*

*}*

到此我们便完成了步骤1的全部工作，即如何在钱包中找到我的可用余额。

我们来继续步骤2：创建tx，把coin放入inputs，把收款人和收款金额放到outpus。

**步骤2：创建tx。**

1. createTransaction方法。

代码非常多，我们下篇来仔细说说吧。