

Bitcoin 比特币官方客户端的高级用法

Bitcoin 比特币官方客户端有两个版本：一个是图形界面的版本，通常被称为 Bitcoin（首字母大写），以及一个简洁的版本（称为 bitcoind）。它们相互间是兼容的，有着同样的命令行参数，读取相同的配置文件，也读写相同的数据文件。您可以在一台电脑中运行 Bitcoin 客户端或是 bitcoind 客户端的其中一个（如果您不小心尝试同时运行另外一个客户端，它会提示您已经有一个客户端在运行并且自动退出）。

命令行参数

使用 `-?` 或 `-help` 参数运行 Bitcoin 或 bitcoind，它会提示常用的命令行参数并退出。

用法：

```
bitcoind [选项]

bitcoind [选项] <命令> [参数] 将命令发送到 -server 或 bitcoind

bitcoind [选项] help          列出命令

bitcoind [选项] help <命令>   获取该命令的帮助
```

选项：

```
-conf=<文件名>    指定配置文件（默认：bitcoin.conf）

-pid=<文件名>     指定 pid（进程 ID）文件（默认：bitcoind.pid）

-gen              生成比特币

-gen=0            不生成比特币

-min              启动时最小化

-splash           启动时显示启动屏幕（默认：1）

-datadir=<目录名> 指定数据目录

-dbcache=<n>      设置数据库缓存大小，单位为兆字节（MB）（默认：25）
```

- dblogsize=<n> 设置数据库磁盘日志大小，单位为兆字节（MB）（默认：100）
- timeout=<n> 设置连接超时，单位为毫秒
- proxy=<ip:端口> 通过 Socks4 代理链接
- dns addnode 允许查询 DNS 并连接
- port=<端口> 监听 <端口> 上的连接（默认：8333，测试网络 testnet：18333）
- maxconnections=<n> 最多维护 <n> 个节点连接（默认：125）
- addnode=<ip> 添加一个节点以供连接，并尝试保持与该节点的连接
- connect=<ip> 仅连接到这里指定的节点
- irc 使用 IRC（因特网中继聊天）查找节点（默认：0）
- listen 接受来自外部的连接（默认：1）
- dnsseed 使用 DNS 查找节点（默认：1）
- banscore=<n> 与行为异常节点断开连接的临界值（默认：100）
- bantime=<n> 重新允许行为异常节点连接所间隔的秒数（默认：86400）
- maxreceivebuffer=<n> 最大每连接接收缓存，<n>*1000 字节（默认：10000）
- maxsendbuffer=<n> 最大每连接发送缓存，<n>*1000 字节（默认：10000）
- upnp 使用全局即插即用（UPNP）映射监听端口（默认：0）
- detachdb 分离货币块和地址数据库。会增加客户端关闭时间（默认：0）
- paytxfee=<amt> 您发送的交易每 KB 字节的手续费
- testnet 使用测试网络
- debug 输出额外的调试信息
- logtimestamps 调试信息前添加时间戳

```
-printtoconsole    发送跟踪/调试信息到控制台而不是 debug.log 文件

-printtodebugger   发送跟踪/调试信息到调试器

-rpcuser=<用户名>  JSON-RPC 连接使用的用户名

-rpcpassword=<密码> JSON-RPC 连接使用的密码

-rpcport=<port>    JSON-RPC 连接所监听的 <端口> (默认: 8332)

-rpcallowip=<ip>   允许来自指定 <ip> 地址的 JSON-RPC 连接

-rpcconnect=<ip>   发送命令到运行在 <ip> 地址的节点 (默认: 127.0.0.1)

-blocknotify=<命令> 当最好的货币块改变时执行命令 (命令中的 %s 会被替换为货币块哈希值)

-upgradewallet     将钱包升级到最新的格式

-keypool=<n>       将密钥池的尺寸设置为 <n> (默认: 100)

-rescan            重新扫描货币块链以查找钱包丢失的交易

-checkblocks=<n>   启动时检查多少货币块 (默认: 2500, 0 表示全部)

-checklevel=<n>    货币块验证的级别 (0-6, 默认: 1)
```

SSL 选项:

```
-rpcssl            使用 OpenSSL (https) JSON-RPC 连接

-rpcsslcertificatechainfile=<文件.cert>  服务器证书文件 (默认: server.cert)

-rpcsslprivatekeyfile=<文件.pem>        服务器私匙文件 (默认: server.pem)

-rpcsslciphers=<密码>                   可接受的密码 (默认: TLSv1+HIGH:!SSLv2:!aNULL:!eNULL:!AH:!3DES:@STRENGTH)
```

bitcoin.conf 配置文件

除了 `-datadir` 和 `-conf` 以外的所有命令行参数都可以通过一个配置文件来设置，而所有配置文件中的选项也都可以在命令行中设置。命令行参数设置的值会覆盖配置文件中的设置。

配置文件是“设置=值”格式的一个列表，每行一个。您还可以使用 `#` 符号来编写注释。

配置文件不会自动创建；您可以使用您喜爱的纯文本编辑器来创建它。默认情况下，Bitcoin（或 bitcoind）会在比特币数据文件夹下查找一个名为“`bitcoin.conf`”的文件，但是数据文件夹和配置文件的路径都可以分别通过 `-datadir` 和 `-conf` 命令行参数分别指定。

操作系统	默认数据文件夹	配置文件路径
Windows	%APPDATA%\Bitcoin\	(XP) C:\Documents and Settings\username\Application Data\Bitcoin\bitcoin.conf (Vista, 7) C:\Users\username\AppData\Roaming\Bitcoin\bitcoin.conf
Linux	\$HOME/.bitcoin/	/home/username/.bitcoin/bitcoin.conf
Mac OS X	\$HOME/Library/Application Support/Bitcoin/	/Users/username/Library/Application Support/Bitcoin/bitcoin.conf

注意：如果 Bitcoin 比特币客户端测试网模式运行，在数据文件夹下客户端会自动创建名为“`testnet`”的子文件夹。

bitcoin.conf 示例

```
# bitcoin.conf 配置文件。以 # 开头的行是注释。
```

```
# 网络相关的设置：
```

```
# 在测试网络中运行，而不是在真正的比特币网络
```

```
#testnet=0
```

通过一个 Socks4 代理服务器连接

#proxy=127.0.0.1:9050

#####

addnode 与 connect 的区别

##

假设您使用了 addnode=4.2.2.4 参数，那么 addnode 便会与

您的节点连接，并且告知您的节点所有与它相连接的其它节点。

另外它还会将您的节点信息告知与其相连接的其它节点，这样它

们也可以连接到您的节点。

##

connect 在您的节点“连接”到它的时候并不会做上述工作。仅

它会与您连接，而其它节点不会。

##

因此如果您位于防火墙后，或者因为其它原因无法找到节点，则

使用“addnode”添加一些节点。

##

如果您想保证隐私，使用“connect”连接到那些您可以“信任”

的节点。

##

如果您在一个局域网内运行了多个节点，您不需要让它们建立许多

连接。您只需要使用“connect”让它们统一连接到一个已端口转

发并拥有多个连接的节点。

#####

您可以在下面使用多个 addnode= 设置来连接到指定的节点

#addnode=69.164.218.197

#addnode=10.0.0.2:8333

... 或使用多个 connect= 设置来仅连接到指定的节点

#connect=69.164.218.197

#connect=10.0.0.1:8333

不使用因特网中继聊天（IRC）（irc.lfnet.org #bitcoin 频道）

来查找其它节点

#noirc=0

入站+出站的最大连接数

#maxconnections=

```
# JSON-RPC 选项（用于控制运行中的 Bitcoin/bitcoind 进程）：

# server=1 告知 Bitcoin-Qt 接受 JSON-RPC 命令

#server=0

# 您必须设置 rpcuser 和 rpcpassword 以确保 JSON-RPC 的安全

#rpcuser=Ulysseys

#rpcpassword=YourSuperGreatPasswordNumber_DO_NOT_USE_THIS_OR_YOU_WILL_GET_ROBBED_385593

# 客户端在 HTTP 连接建立后，等待多少秒以完成一个 RPC HTTP 请求

#rpctimeout=30

# 默认仅允许来自本机的 RPC 连接。在这里您可以指定多个

# rpcallowip=, 来设置您想允许连接的其它主机 IP 地址。

# 您可以使用 * 作为通配符。

#rpcallowip=10.1.1.34

#rpcallowip=192.168.1.*

# 在如下端口监听 RPC 连接

#rpcport=8332
```

您可以通过如下设置使用 Bitcoin 或 bitcoind 来发送命令到一个在

其它主机远程运行的 Bitcoin/bitcoind 客户端

#rpcconnect=127.0.0.1

使用安全套接层（也称为 TLS 或 HTTPS）来

连接到 Bitcoin -server 或 bitcoind

#rpcssl=1

当 rpcssl=1 时使用的 OpenSSL 设置

#rpcsslcipher=TLSv1+HIGH:!SSLv2:!aNULL:!eNULL:!AH:!3DES:@STRENGTH

#rpcsslcertificatechainfile=server.cert

#rpcsslprivatekeyfile=server.pem

其它选项:

设置 gen=1 以尝试生成比特币（采矿）

#gen=0

预生成如下数目的公匙和私匙，这样钱包备份便可以对已有的交易以及未来

多笔交易有效

#keypool=100

每次您发送比特币的时候支付一个可选的额外的交易手续费。包含手续费的交易

会更快的被包含在新生成的货币块中，因此会更快生效

#paytxfee=0.00

允许直接连接，实现“通过 IP 地址支付”功能

#allowreceivebyip=1

用户界面选项:

最小化启动比特币客户端

#min=1

最小化到系统托盘

#minimizetotray=1