

# 以太坊客户端 Geth 命令用法-参数详解

作者：守护平井一夫【链客】

Geth 在以太坊智能合约开发中最常用的工具（必备开发工具），一个多用途的命令行工具。

熟悉 Geth 可以让我们有更好的效率，大家可收藏起来作为 Geth 命令用法手册。本文主

要是 `geth help` 的翻译，基于最新的 `geth 1.7.3-stable` 版本。

以下开始正文。

## 命令用法

```
geth [选项] 命令 [命令选项] [参数...]
```

## 版本：

```
1.7.3-stable
```

## 命令：

<code>account</code>	管理账户
<code>attach</code>	启动交互式 JavaScript 环境（连接到节点）
<code>bug</code>	上报 bug Issues
<code>console</code>	启动交互式 JavaScript 环境
<code>copydb</code>	从文件夹创建本地链
<code>dump</code>	Dump（分析）一个特定的块存储
<code>dumpconfig</code>	显示配置值
<code>export</code>	导出区块链到文件

<code>import</code>	导入一个区块链文件
<code>init</code>	启动并初始化一个新的创世块
<code>js</code>	执行指定的 <code>JavaScript</code> 文件(多个)
<code>license</code>	显示许可信息
<code>makecache</code>	生成 <code>ethash</code> 验证缓存(用于测试)
<code>makedag</code>	生成 <code>ethash</code> 挖矿 <code>DAG</code> (用于测试)
<code>monitor</code>	监控和可视化节点指标
<code>removedb</code>	删除区块链和状态数据库
<code>version</code>	打印版本号
<code>wallet</code>	管理 <code>Ethereum</code> 预售钱包
<code>help,h</code>	显示一个命令或帮助一个命令列表

## ETHEREUM 选项:

<code>--config value</code>	<code>TOML</code> 配置文件
<code>--datadir "xxx"</code>	数据库和 <code>keystore</code> 密钥的数据目录
<code>--keystore</code>	<code>keystore</code> 存放目录(默认在 <code>datadir</code> 内)
<code>--nousb</code>	禁用监控和管理 <code>USB</code> 硬件钱包
<code>--networkid value</code>	网络标识符(整型, 1= <code>Frontier</code> , 2= <code>Morden</code> (弃用), 3= <code>Ropsten</code> , 4= <code>Rinkeby</code> ) (默认: 1)
<code>--testnet</code>	<code>Ropsten</code> 网络: 预先配置的 <code>POW</code> ( <code>proof-of-work</code> )测试网络
<code>--rinkeby</code>	<code>Rinkeby</code> 网络: 预先配置的 <code>POA</code> ( <code>proof-of-authority</code> )测试网络
<code>--syncmode "fast"</code>	同步模式 (" <code>fast</code> ", " <code>full</code> ", or " <code>light</code> ")
<code>--ethstats value</code>	上报 <code>ethstats</code> service URL ( <code>nodename:secret@host:port</code> )
<code>--identity value</code>	自定义节点名

<code>--lightserv value</code>	允许 LES 请求时间最大百分比(0 - 90)(默认值:0)
<code>--lightpeers value</code>	最大 LES client peers 数量(默认值:20)
<code>--lightkdf</code>	在 KDF 强度消费时降低 key-derivation RAM&CPU 使用

## 开发者 (模式) 选项:

<code>--dev</code>	使用 POA 共识网络, 默认预分配一个开发者账户并且会自动开启挖矿。
<code>--dev.period value</code>	开发者模式下挖矿周期 (0 = 仅在交易时) (默认: 0)

## ETHASH 选项:

<code>--ethash.cachedir</code>	ethash 验证缓存目录(默认 = datadir 目录内)
<code>--ethash.cachesinmem value</code>	在内存保存的最近的 ethash 缓存个数 (每个缓存 16MB) (默认: 2)
<code>--ethash.cachesondisk value</code>	在磁盘保存的最近的 ethash 缓存个数 (每个缓存 16MB) (默认: 3)
<code>--ethash.dagdir ""</code>	存 ethash DAGs 目录 (默认 = 用户 hom 目录)
<code>--ethash.dagsinmem value</code>	在内存保存的最近的 ethash DAGs 个数 (每个 1GB 以上) (默认: 1)
<code>--ethash.dagsondisk value</code>	在磁盘保存的最近的 ethash DAGs 个数 (每个 1GB 以上) (默认: 2)

## 交易池选项:

<code>--txpool.nolocals</code>	为本地提交交易禁用价格豁免
<code>--txpool.journal value</code>	本地交易的磁盘日志: 用于节点重启 (默认: "transactions.rlp")
<code>--txpool.rejournal value</code>	重新生成本地交易日志的时间间隔 (默认: 1 小时)
<code>--txpool.pricelimit value</code>	加入交易池的最小的 gas 价格限制(默认: 1)
<code>--txpool.pricebump value</code>	价格波动百分比 (相对之前已有交易) (默认: 10)

--txpool.accountslots value 每个帐户保证可执行的最少交易槽数量 (默认: 16)

--txpool.globalslots value 所有帐户可执行的最大交易槽数量 (默认: 4096)

--txpool.accountqueue value 每个帐户允许的最多非可执行交易槽数量 (默认: 64)

--txpool.globalqueue value 所有帐户非可执行交易最大槽数量 (默认: 1024)

--txpool.lifetime value 非可执行交易最大入队时间(默认: 3 小时)

## 性能调优的选项:

--cache value 分配给内部缓存的内存 MB 数量, 缓存值(最低 16 mb / 数据库强制要求)(默认:128)

--trie-cache-gens value 保持在内存中产生的 trie node 数量(默认:120)

## 帐户选项:

--unlock value 需解锁账户用逗号分隔

--password value 用于非交互式密码输入的密码文件

## API 和控制台选项:

--rpc 启用 HTTP-RPC 服务器

--rpcaddr value HTTP-RPC 服务器接口地址(默认值:"localhost")

--rpcport value HTTP-RPC 服务器监听端口(默认值:8545)

--rpcapi value 基于 HTTP-RPC 接口提供的 API

--ws 启用 WS-RPC 服务器

--wsaddr value WS-RPC 服务器监听接口地址(默认值:"localhost")

--wsport value WS-RPC 服务器监听端口(默认值:8546)

--wsapi value 基于 WS-RPC 的接口提供的 API

--wsorigins value websockets 请求允许的源

<code>--ipcdisable</code>	禁用 IPC-RPC 服务器
<code>--ipcpath</code>	包含在 <code>datadir</code> 里的 IPC <code>socket/pipe</code> 文件名(转义过的显式路径)
<code>--rpccorsdomain value</code>	允许跨域请求的域名列表(逗号分隔)(浏览器强制)
<code>--jspath loadScript</code>	JavaScript 加载脚本的根路径(默认值:“.”)
<code>--exec value</code>	执行 JavaScript 语句(只能结合 <code>console/attach</code> 使用)
<code>--preload value</code>	预加载到控制台的 JavaScript 文件列表(逗号分隔)

## 网络选项:

<code>--bootnodes value</code>	用于 P2P 发现引导的 <code>enode urls</code> (逗号分隔)(对于 <code>light servers</code> 用 <code>v4+v5</code> 代替)
<code>--bootnodesv4 value</code>	用于 P2P v4 发现引导的 <code>enode urls</code> (逗号分隔) ( <code>light server</code> , 全节点)
<code>--bootnodesv5 value</code>	用于 P2P v5 发现引导的 <code>enode urls</code> (逗号分隔) ( <code>light server</code> , 轻节点)
<code>--port value</code>	网卡监听端口(默认值:30303)
<code>--maxpeers value</code>	最大的网络节点数量(如果设置为 0, 网络将被禁用)(默认值:25)
<code>--maxpendpeers value</code>	最大尝试连接的数量(如果设置为 0, 则将使用默认值)(默认值:0)
<code>--nat value</code>	NAT 端口映射机制 ( <code>any none upnp pmp extip:&lt;IP&gt;</code> ) (默认:“any”)
<code>--nodiscover</code>	禁用节点发现机制(手动添加节点)
<code>--v5disc</code>	启用实验性的 RLPx V5(Topic 发现)机制
<code>--nodekey value</code>	P2P 节点密钥文件
<code>--nodekeyhex value</code>	十六进制的 P2P 节点密钥(用于测试)

## 矿工选项:

```
--mine                打开挖矿

--minerthreads value  挖矿使用的 CPU 线程数量(默认值:8)

--etherbase value     挖矿奖励地址(默认=第一个创建的帐户)(默认值:"0")

--targetgaslimit value 目标 gas 限制: 设置最低 gas 限制 (低于这个不会被挖?) (默认值:"4712388")

--gasprice value      挖矿接受交易的最低 gas 价格

--extradata value     矿工设置的额外块数据(默认=client version)
```

## GAS 价格选项:

```
--gpoblocks value    用于检查 gas 价格的最近块的个数 (默认: 10)

--gpopercentile value 建议 gas 价参考最近交易的 gas 价的百分位数, (默认: 50)
```

## 虚拟机的选项:

```
--vmdebug            记录 VM 及合约调试信息
```

## 日志和调试选项:

```
--metrics            启用 metrics 收集和报告

--fakepow            禁用 proof-of-work 验证

--verbosity value    日志详细度:0=silent, 1=error, 2=warn, 3=info, 4=debug, 5=detail (default: 3)

--vmodule value      每个模块详细度:以 <pattern>=<level>的逗号分隔列表 (比如 eth/*=6,p2p=5)

--backtrace value    请求特定日志记录堆栈跟踪 (比如 "block.go:271")

--debug              突出显示调用位置日志(文件名及行号)

--pprof              启用 pprof HTTP 服务器

--pprofaddr value    pprof HTTP 服务器监听接口(默认值:127.0.0.1)
```

--pprofport value	pprof HTTP 服务器监听端口(默认值:6060)
--memprofilerate value	按指定频率打开 memory profiling (默认:524288)
--blockprofilerate value	按指定频率打开 block profiling (默认值:0)
--cpuprofile value	将 CPU profile 写入指定文件
--trace value	将 execution trace 写入指定文件

## WHISPER 实验选项:

--shh	启用 Whisper
--shh.maxmessagesize value	可接受的最大的消息大小 (默认值: 1048576)
--shh.pow value	可接受的最小的 POW (默认值: 0.2)

## 弃用选项:

--fast	开启快速同步
--light	启用轻客户端模式

## 其他选项:

-help, -h	显示帮助
-----------	------

我是【链客】六级算力等级《守护平井一夫》 为各位解答区块链技术问题，欢迎加入。

链客区块链技术问答社区，有问必答！！

国内域名: [www.liankexing.com](http://www.liankexing.com)

复制网址至浏览器即可进入社区

国际域名: [www.lk.wiki](http://www.lk.wiki)

QQ 群: 725414372

