peer channel

`peer channel` 命令允许管理员在 Peer 上执行通道相关的操作，比如加入通道，或者列出当前 Peer 加入的通道。

语法

`peer channel` 命令有以下子命令：

- create
- fetch
- getinfo
- join
- list
- signconfigtx
- update

peer channel

```
Operate a channel: create|fetch|join|list|update|signconfigtx|getinfo.

Usage:
  peer channel [command]

Available Commands:
  create       Create a channel
  fetch        Fetch a block
  getinfo      get blockchain information of a specified channel.
  join         Joins the peer to a channel.
  list         List of channels peer has joined.
  signconfigtx Signs a configtx update.
  update       Send a configtx update.

Flags:
      --cafile string                       Path to file containing PEM-encoded trusted certifica
      --certfile string                     Path to file containing PEM-encoded X509 public key t
      --clientauth                          Use mutual TLS when communicating with the orderer en
      --connTimeout duration                Timeout for client to connect (default 3s)
  -h, --help                                help for channel
      --keyfile string                      Path to file containing PEM-encoded private key to us
  -o, --orderer string                      Ordering service endpoint
      --ordererTLSHostnameOverride string   The hostname override to use when validating the TLS
      --tls                                 Use TLS when communicating with the orderer endpoint

Use "peer channel [command] --help" for more information about a command.
```

peer channel create

```
Create a channel and write the genesis block to a file.

Usage:
  peer channel create [flags]

Flags:
  -c, --channelID string     In case of a newChain command, the channel ID to create. It must be
  -f, --file string          Configuration transaction file generated by a tool such as configtxg
  -h, --help                 help for create
```

```
      --outputBlock string   The path to write the genesis block for the channel. (default ./<cha
  -t, --timeout duration     Channel creation timeout (default 10s)

Global Flags:
      --cafile string                       Path to file containing PEM-encoded trusted certifica
      --certfile string                     Path to file containing PEM-encoded X509 public key t
      --clientauth                          Use mutual TLS when communicating with the orderer en
      --connTimeout duration                Timeout for client to connect (default 3s)
      --keyfile string                      Path to file containing PEM-encoded private key to us
  -o, --orderer string                      Ordering service endpoint
      --ordererTLSHostnameOverride string   The hostname override to use when validating the TLS
      --tls                                 Use TLS when communicating with the orderer endpoint
```

peer channel fetch

```
Fetch a specified block, writing it to a file.

Usage:
  peer channel fetch <newest|oldest|config|(number)> [outputfile] [flags]

Flags:
      --bestEffort         Whether fetch requests should ignore errors and return blocks on a bes
  -c, --channelID string   In case of a newChain command, the channel ID to create. It must be al
  -h, --help               help for fetch

Global Flags:
      --cafile string                       Path to file containing PEM-encoded trusted certifica
      --certfile string                     Path to file containing PEM-encoded X509 public key t
      --clientauth                          Use mutual TLS when communicating with the orderer en
      --connTimeout duration                Timeout for client to connect (default 3s)
      --keyfile string                      Path to file containing PEM-encoded private key to us
  -o, --orderer string                      Ordering service endpoint
      --ordererTLSHostnameOverride string   The hostname override to use when validating the TLS
      --tls                                 Use TLS when communicating with the orderer endpoint
```

peer channel getinfo

```
get blockchain information of a specified channel. Requires '-c'.

Usage:
  peer channel getinfo [flags]

Flags:
  -c, --channelID string   In case of a newChain command, the channel ID to create. It must be al
  -h, --help               help for getinfo

Global Flags:
      --cafile string                       Path to file containing PEM-encoded trusted certifica
      --certfile string                     Path to file containing PEM-encoded X509 public key t
      --clientauth                          Use mutual TLS when communicating with the orderer en
      --connTimeout duration                Timeout for client to connect (default 3s)
      --keyfile string                      Path to file containing PEM-encoded private key to us
  -o, --orderer string                      Ordering service endpoint
      --ordererTLSHostnameOverride string   The hostname override to use when validating the TLS
      --tls                                 Use TLS when communicating with the orderer endpoint
```

peer channel join

```
Joins the peer to a channel.

Usage:
  peer channel join [flags]

Flags:
  -b, --blockpath string   Path to file containing genesis block
  -h, --help               help for join
```

```
Global Flags:
      --cafile string                       Path to file containing PEM-encoded trusted certifica
      --certfile string                     Path to file containing PEM-encoded X509 public key t
      --clientauth                          Use mutual TLS when communicating with the orderer en
      --connTimeout duration                Timeout for client to connect (default 3s)
      --keyfile string                      Path to file containing PEM-encoded private key to us
  -o, --orderer string                      Ordering service endpoint
      --ordererTLSHostnameOverride string   The hostname override to use when validating the TLS
      --tls                                 Use TLS when communicating with the orderer endpoint
```

peer channel list

```
List of channels peer has joined.

Usage:
  peer channel list [flags]

Flags:
  -h, --help   help for list

Global Flags:
      --cafile string                       Path to file containing PEM-encoded trusted certifica
      --certfile string                     Path to file containing PEM-encoded X509 public key t
      --clientauth                          Use mutual TLS when communicating with the orderer en
      --connTimeout duration                Timeout for client to connect (default 3s)
      --keyfile string                      Path to file containing PEM-encoded private key to us
  -o, --orderer string                      Ordering service endpoint
      --ordererTLSHostnameOverride string   The hostname override to use when validating the TLS
      --tls                                 Use TLS when communicating with the orderer endpoint
```

peer channel signconfigtx

```
Signs the supplied configtx update file in place on the filesystem. Requires '-f'.

Usage:
  peer channel signconfigtx [flags]

Flags:
  -f, --file string   Configuration transaction file generated by a tool such as configtxgen for
  -h, --help          help for signconfigtx

Global Flags:
      --cafile string                       Path to file containing PEM-encoded trusted certifica
      --certfile string                     Path to file containing PEM-encoded X509 public key t
      --clientauth                          Use mutual TLS when communicating with the orderer en
      --connTimeout duration                Timeout for client to connect (default 3s)
      --keyfile string                      Path to file containing PEM-encoded private key to us
  -o, --orderer string                      Ordering service endpoint
      --ordererTLSHostnameOverride string   The hostname override to use when validating the TLS
      --tls                                 Use TLS when communicating with the orderer endpoint
```

peer channel update

```
Signs and sends the supplied configtx update file to the channel. Requires '-f', '-o', '-c'.

Usage:
  peer channel update [flags]

Flags:
  -c, --channelID string   In case of a newChain command, the channel ID to create. It must be al
  -f, --file string        Configuration transaction file generated by a tool such as configtxgen
  -h, --help               help for update

Global Flags:
      --cafile string                       Path to file containing PEM-encoded trusted certifica
      --certfile string                     Path to file containing PEM-encoded X509 public key t
      --clientauth                          Use mutual TLS when communicating with the orderer en
```

```
      --connTimeout duration                Timeout for client to connect (default 3s)
      --keyfile string                      Path to file containing PEM-encoded private key to us
  -o, --orderer string                      Ordering service endpoint
      --ordererTLSHostnameOverride string   The hostname override to use when validating the TLS
      --tls                                 Use TLS when communicating with the orderer endpoint
```

使用示例

peer channel create 示例

本样例展示了 `peer channel create` 使用全局标识 `--orderer` 的用法。

- 使用 `./createchannel.tx` 中的配置交易创建样例通道 `mychannel`。使用排序节点 `orderer.example.com:7050`。

  ```
  peer channel create -c mychannel -f ./createchannel.tx --orderer orderer.example.com:7050

  2018-02-25 08:23:57.548 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer conne
  2018-02-25 08:23:57.626 UTC [channelCmd] InitCmdFactory -> INFO 019 Endorser and orderer conne
  2018-02-25 08:23:57.834 UTC [channelCmd] readBlock -> INFO 020 Received block: 0
  2018-02-25 08:23:57.835 UTC [main] main -> INFO 021 Exiting.....
  ```

  返回区块 0 代表已经成功创建通道。

下一个例子展示使用 `peer channel create` 的命令选项。

- 使用 `orderer.example.com:7050` 创建新的通道 `mychannel`，配置交易同样定义在 `./createchannel.tx` 文件中。但多了通道创建等待30s的选项。

  ```
  peer channel create -c mychannel --orderer orderer.example.com:7050 -f ./createchannel.tx -t

  2018-02-23 06:31:58.568 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer con
  2018-02-23 06:31:58.669 UTC [channelCmd] InitCmdFactory -> INFO 019 Endorser and orderer con
  2018-02-23 06:31:58.877 UTC [channelCmd] readBlock -> INFO 020 Received block: 0
  2018-02-23 06:31:58.878 UTC [main] main -> INFO 021 Exiting.....

  ls -l

  -rw-r--r-- 1 root root 11982 Feb 25 12:24 mychannel.block
  ```

  你可以看到输出了区块0，证明了 `mychannel` 创建成功了，区块0存在了本地目录，名字为 `mychanenl.block`。

  区块0通常被长尾 *创世块*，因为它包含了通道的初始配置。所有对通道的更新，都会创建配置区块存在通道的区块链上，并且新配置区块中的配置会取代老的配置。

peer channel fetch 样例

以下是 `peer channel fetch` 命令的样例.

- 使用 `newest` 选项获取指定通道的最新区块，并把区块保存到 `mychanenl.block` 文件中。

  ```
  peer channel fetch newest mychannel.block -c mychannel --orderer orderer.example.com:7050
  ```

```
2018-02-25 13:10:16.137 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer conne
2018-02-25 13:10:16.144 UTC [channelCmd] readBlock -> INFO 00a Received block: 32
2018-02-25 13:10:16.145 UTC [main] main -> INFO 00b Exiting.....

ls -l

-rw-r--r-- 1 root root 11982 Feb 25 13:10 mychannel.block
```

你可以看到获取的区块高度是32，并且区块已经被写入到 `mychanenl.block` 文件中。

- 使用 `(block number)` 获取指定的区块，并且保存到默认的区块文件，本例中区块号是16。

```
peer channel fetch 16  -c mychannel --orderer orderer.example.com:7050

2018-02-25 13:46:50.296 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer conne
2018-02-25 13:46:50.302 UTC [channelCmd] readBlock -> INFO 00a Received block: 16
2018-02-25 13:46:50.302 UTC [main] main -> INFO 00b Exiting.....

ls -l

-rw-r--r-- 1 root root 11982 Feb 25 13:10 mychannel.block
-rw-r--r-- 1 root root  4783 Feb 25 13:46 mychannel_16.block
```

你可以看到获取的区块高度是16，并且区块已经被写入到 `mychanenl.block` 文件中。

对于配置区块，可以使用 `configtxlator` 命令解析区块文件。请查看该命令的帮助信息获取解析样例。用户交易区块同样可以被解析，但需要写一个专门的程序做这件事。

peer channel getinfo example

如下是 `peer channel getinfo` 命令的使用样例。

- 获取当前 Peer 节点上 `mychannel` 通道的信息。

```
peer channel getinfo -c mychannel

2018-02-25 15:15:44.135 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer conne
Blockchain info: {"height":5,"currentBlockHash":"JgK9lcaPUNmFb5Mp1qe1SVMsx3o/22Ct4+n5tejcXCw="
2018-02-25 15:15:44.139 UTC [main] main -> INFO 006 Exiting.....
```

你可以看到 `mychannel` 最新的区块是5，以及当前通道中，最近区块的加密哈希值。

peer channel join example

如下是 `peer channel join` 命令的例子.

- 把一个 Peer 加入到 `./mychannel.genesis.block` 定义的通道。本例中，通道配置块是之前通过 `peer channel fetch` 命令获取的区块。

```
peer channel join -b ./mychannel.genesis.block

2018-02-25 12:25:26.511 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer conne
2018-02-25 12:25:26.571 UTC [channelCmd] executeJoin -> INFO 006 Successfully submitted propos
2018-02-25 12:25:26.571 UTC [main] main -> INFO 007 Exiting.....
```

你可以看到 Peer 已成功创建了加入通道的交易。

peer channel list example

如下是 `peer channel list` 命令的样例。

- 列出 Peer 加入的通道。

```
peer channel list

2018-02-25 14:21:20.361 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer conne
Channels peers has joined:
mychannel
2018-02-25 14:21:20.372 UTC [main] main -> INFO 006 Exiting.....
```

  你可以看到 Peer 加入了 `mychannel` 通道.

peer channel signconfigtx example

如下是 `peer channel signconfigtx` 命令的样例。

- 为定义在 `./updatechannel.txn` 中的 `channel update` 交易进行签名。样例在执行命令前后列出了配置交易文件。

```
ls -l

-rw-r--r--  1 anthonyodowd  staff   284 25 Feb 18:16 updatechannel.tx

peer channel signconfigtx -f updatechannel.tx

2018-02-25 18:16:44.456 GMT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer conne
2018-02-25 18:16:44.459 GMT [main] main -> INFO 002 Exiting.....

ls -l

-rw-r--r--  1 anthonyodowd  staff  2180 25 Feb 18:16 updatechannel.tx
```

  你可以看到配置交易文件 `updatechannel.tx` 的大小从 284 字节增加到 2180 字节，说明 Peer 成功对配置交易文件进行了签名。

peer channel update example

如下是 `peer channel update` 命令的样例.

- 使用 `./updatechannel.tx` 中定义的配置交易更新 `mychannel` 的配置。使用 `orderer.example.com:7050` 作为排序节点，把配置交易发送给在通道中的所有 Peer，让它们更新本地通道的配置。

```
peer channel update -c mychannel -f ./updatechannel.tx -o orderer.example.com:7050

2018-02-23 06:32:11.569 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer conne
2018-02-23 06:32:11.626 UTC [main] main -> INFO 010 Exiting.....
```

  可以看到通道 `mychannel` 成功被更新。