

configtx.yaml是Hyperledger Fabric区块链网络运维工具configtxgen用于生成通道创世块或通道交易的配置文件，configtx.yaml的内容直接决定了所生成的创世区块的内容

参考：

<https://github.com/hyperledger/fabric/blob/master/sampleconfig/configtx.yaml>

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

---
#####
#####
#
#   Section: Organizations
#
#   - 本节定义了不同的组织标识，这些标识将在稍后的配置中引用。
#
#####
#####
Organizations:

    # SampleOrg使用sampleconfig定义了一个MSP。它不应该在生产中使用，但可以
    用作其他定义的模板
    - &OrdererOrg
        # DefaultOrg定义了使用在fabric.git开发环境的sampleconfig中的组织
        Name: OrdererOrg

        # ID 下载MSP的ID
        ID: OrdererMSP

        # MSPDir是包含MSP配置的文件系统路径，由cryptogen工具生成的加密材料路
        径
        MSPDir: crypto-config/ordererOrganizations/example.com/msp

        # Policies定义了这个配置树级别的策略集
        # 对于组织策略，它们的规范路径通常是/Channel/<Application|Orderer>/<OrgName>/<PolicyName>
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Writers:
                Type: Signature
```

```

        Rule: "OR('OrdererMSP.member')"
    Admins:
        Type: Signature
        Rule: "OR('OrdererMSP.admin')"

- &Org1
    # DefaultOrg定义了使用在fabric.git开发环境的sampleconfig中的组织
    Name: Org1MSP

    # ID 下载MSP的ID
    ID: Org1MSP

    MSPDir: crypto-
config/peerOrganizations/org1.example.com/msp

    # Policies定义了在这个配置树级别的策略集
    # 对于组织策略，它们的规范路径通常是/Channel/<Application|Orderer>/<OrgName>/<PolicyName>
    Policies:
        Readers:
            Type: Signature
            Rule: "OR('Org1MSP.admin', 'Org1MSP.peer',
'Org1MSP.client')"
        Writers:
            Type: Signature
            Rule: "OR('Org1MSP.admin', 'Org1MSP.client')"
        Admins:
            Type: Signature
            Rule: "OR('Org1MSP.admin')"

    # 将此标志设置为true.
    AnchorPeers: #指明org1中使用peer0作为锚节点
        # AnchorPeers定义了可以用于跨组织gossip通信的节点的位置
        # 注意，这个值只在应用程序部分上下文的genesis块中编码
        - Host: peer0.org1.example.com
          Port: 7051

- &Org2
    # DefaultOrg定义了使用在fabric.git开发环境的sampleconfig中的组织
    Name: Org2MSP

    # ID 来指定下载的MSP定义
    ID: Org2MSP

    MSPDir: crypto-
config/peerOrganizations/org2.example.com/msp

```

```

# Policies定义了在这个配置树级别的策略集
# 对于组织策略，它们的规范路径通常是/Channel/<Application|Orderer>/<OrgName>/<PolicyName>
Policies:
  Readers:
    Type: Signature
    Rule: "OR('Org2MSP.admin', 'Org2MSP.peer', 'Org2MSP.client')"
  Writers:
    Type: Signature
    Rule: "OR('Org2MSP.admin', 'Org2MSP.client')"
  Admins:
    Type: Signature
    Rule: "OR('Org2MSP.admin')"

AnchorPeers: #指明org2中使用peer0作为锚节点
# AnchorPeers定义了可以用于跨组织gossip通信的节点的位置
# 注意，这个值只在应用程序部分上下文的genesis块中编码
- Host: peer0.org2.example.com
  Port: 9051

#####
#####
#
# SECTION: Capabilities
#
# - 本节定义fabric network的功能。这是v1.1.0的一个新概念，不应该在带有v1.0.x版本orderers和peers的混合网络中使用。
# 功能定义了fabric二进制文件中必须提供的特性，以便该二进制文件安全地参与fabric网络。
# 例如，如果添加了新的MSP类型，较新的二进制文件可能会识别并验证来自该类型的签名，而没有此支持的较老的二进制文件将无法验证这些交易。
# 这可能导致不同版本的fabric二进制文件具有不同的世界状态。相反，为通道定义一个功能会通知那些没有这个功能的二进制文件，它们必须停止处理交易，直到它们被升级。
# 对于v1.0.x 版本，如果定义了任何功能(包括关闭所有功能的映射)，v1.0.x版本的节点可能会故意崩溃。
#
#####
#####
Capabilities:
# 通道功能同时适用于orderers和peers，并且必须得到双方的支持。
# 将功能的值设置为true以满足需要。
Channel: &ChannelCapabilities
# Channel的V1.3是一个用来表示运行在V1.3.x版本上的所有orderers和peers都需要的行为的catchall标志，但这将与以前版本中的orderers和peers不兼容。
# 在启用V1.3通道功能之前，请确保通道上的所有orderers和peers都位于

```

v1.3.0或更高版本。

```
V1_3: true
```

Orderer功能只适用于Orderer，并且可以安全地与以前版本的节点一起使用。

将功能的值设置为true以满足需要

```
Orderer: &OrdererCapabilities
```

Orderer的V1.1是一个用来表示确定被在v1.1.x中运行的所有orderers所需要的行为的catchall标志，但这与以前版本中的orderers不兼容。

在启用V1.1 orderer功能之前，请确保通道上的所有orderer都处于v1.1.0或更高版本。

```
V1_1: true
```

Application功能只应用于peer网络，并且可以安全地与以前的版本的orderers一起使用。

将功能的值设置为true以满足需要

```
Application: &ApplicationCapabilities
```

Application的V1.3用来启用新的非向后兼容特性fabric V1.3的补丁V1_3: true

Application的V1.2用来启用新的非向后兼容特性和fabric V1.2的补丁(注意，如果设置了之后的版本的功能，则不需要设置此功能)

```
V1_2: false
```

Application的V1.1启用新的非向后兼容特性和fabric V1.1的补丁(注意，如果设置了之后的版本的功能，则不需要设置此特性)。

所以这里设置是V1_3为true，另外两个为false

```
V1_1: false
```

```
#####  
#####
```

```
#
```

```
# SECTION: Application
```

```
#
```

```
# - 本节为Application相关参数定义要编码到配置交易或创世区块中的值
```

```
# 注意，只有在创建channel tx时才会用Application，否则是创建区块
```

```
#
```

```
#####  
#####
```

```
Application: &ApplicationDefaults
```

Organizations是定义为网络应用程序端参与者的组织列表

```
Organizations:
```

Policies定义了这个配置树级别的策略集

对于组织策略，它们的规范路径通常是/Channel/<Application|Orderer>/<OrgName>/<PolicyName>

```
Policies:
```

```
Readers:
```

```
  Type: ImplicitMeta
```

```

        Rule: "ANY Readers"

Writers:
    Type: ImplicitMeta
    Rule: "ANY Writers"
Admins:
    Type: ImplicitMeta
    Rule: "MAJORITY Admins"

Capabilities:
    <<: *ApplicationCapabilities
#####
#####
#
#   SECTION: Orderer
#
#   - 本节为orderer相关参数定义要编码到配置交易或创世区块中的值
#
#####
#####
Orderer: &OrdererDefaults

# Orderer 类型：要启动的Orderer实现类型
# 可用的类型有“solo”和“kafka”
OrdererType: solo

Addresses:
    - orderer.example.com:7050

# Batch Timeout: 创建批处理之前要等待的时间
BatchTimeout: 2s

# Batch Size: 控制成块的消息数量
BatchSize:

# Max Message Count: 批处理中允许的最大消息数
MaxMessageCount: 10

# Absolute Max Bytes: 批处理中允许序列化消息的绝对最大字节数。
AbsoluteMaxBytes: 99 MB

# Preferred Max Bytes: 批处理中允许序列化消息的首选最大字节数。
# 大于首选最大字节的消息将导致批处理大于改值。
PreferredMaxBytes: 512 KB

Kafka:
# Brokers: orderer连接到的Kafka代理的列表
# NOTE: Use IP:port notation
Brokers:

```

- 127.0.0.1:9092

Organizations是组织的列表，组织被定义为网络的orderer方的参与者
Organizations:

Policies定义了这个配置树级别的策略集

对于Orderer策略，它们的规范路径通常是/Channel/Orderer/<PolicyName>
Policies:

Readers:

Type: ImplicitMeta

Rule: "ANY Readers"

Writers:

Type: ImplicitMeta

Rule: "ANY Writers"

Admins:

Type: ImplicitMeta

Rule: "MAJORITY Admins"

BlockValidation指定必须包含来自orderer的哪些签名，以便peer对其进行验证。

BlockValidation:

Type: ImplicitMeta

Rule: "ANY Writers"

#####

#

CHANNEL

#

本节为channel相关参数定义要编码到配置交易或创世区块中的值

#

#####

Channel: &ChannelDefaults

Policies定义了这个配置树级别的策略集

对于Channel策略，它们的规范路径通常是/Channel/<PolicyName>

Policies:

谁可能调用 'Deliver' API

Readers:

Type: ImplicitMeta

Rule: "ANY Readers"

谁可能调用 'Broadcast' API

Writers:

Type: ImplicitMeta

Rule: "ANY Writers"

默认，谁可能调用在这个配置级别的元素

Admins:

Type: ImplicitMeta

Rule: "MAJORITY Admins"

Capabilities描述通道级功能，有关详细描述，请参阅此文件中其他部分的专用功能部分

Capabilities:

<<: *ChannelCapabilities

#####

#

Profile

#

- 这里可以编码不同的配置概要文件，将其指定为configtxgen工具的参数

#

#####

Profiles:

TwoOrgsOrdererGenesis:

<<: *ChannelDefaults

Orderer:

<<: *OrdererDefaults

Organizations:

- *OrdererOrg

Capabilities:

<<: *OrdererCapabilities

Consortiums:

SampleConsortium:

Organizations:

- *Org1

- *Org2

TwoOrgsChannel:

Consortium: SampleConsortium

Application:

<<: *ApplicationDefaults

Organizations:

- *Org1

- *Org2

Capabilities:

<<: *ApplicationCapabilities

SampleDevModeKafka:

<<: *ChannelDefaults

Capabilities:

<<: *ChannelCapabilities

Orderer:

<<: *OrdererDefaults

```
OrdererType: kafka
Kafka:
  Brokers:
    - kafka.example.com:9092

Organizations:
  - *OrdererOrg
Capabilities:
  <<: *OrdererCapabilities
Application:
  <<: *ApplicationDefaults
Organizations:
  - <<: *OrdererOrg
Consortiums:
  SampleConsortium:
    Organizations:
      - *Org1
      - *Org2
```