MultiChain Simulation Project – Documentation

Contents

Simulation Program

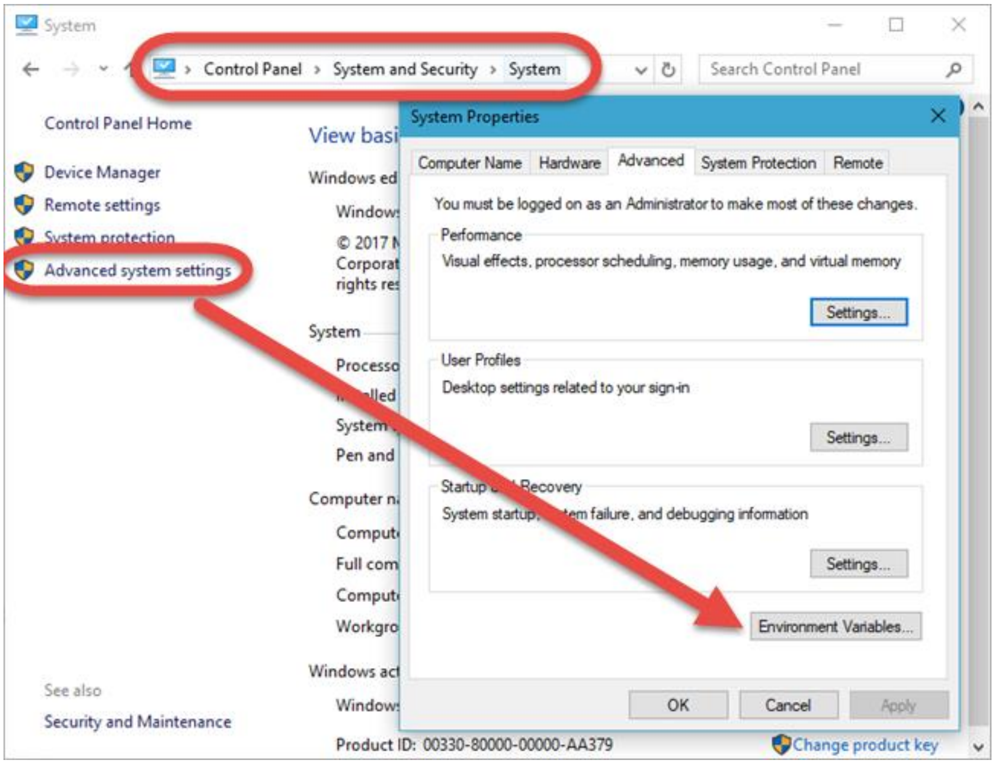
Prerequisites

In order to get started and run the Java project, you will need the following:

1. A Java development kit, with the most recent version available for download under https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html
2. An integrated development environment (IDE) such as Eclipse, available for download under https://www.eclipse.org/downloads/.

Installation

1. Download the MultiChain executables (preferably the latest version of 1.0, or the Beta version of 2.0 if you wish) from https://www.multichain.com/download-install/.
2. Unzip to desired folder.
3. Import the project containing the simulation project to an IDE of your choice.
4. Create two environmental variables named “MULTICHAIN” containing the path to the folder with the MultiChain executables and “CHAINDATA” linking to the folder where the chain data is stored (usually C:\Users\<User>\AppData\Roaming\MultiChain).



You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

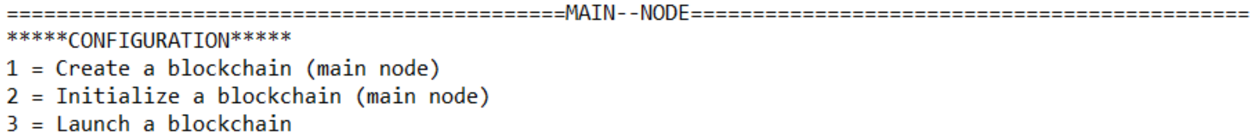
Product ID: 00330-80000-00000-AA379

5. If needed, download or update the JAR files for the MultiChain Java API from https://github.com/SimplyUb/MultiChainJavaAPI. The project was tested with version “0.5.01-SNAPSHOT”. Make sure to use the JAR file with dependencies and add it to your project.

Note: If you are trying to access a blockchain or node running on a different machine with a specific address, the installation of the MultiChain executables is not required.

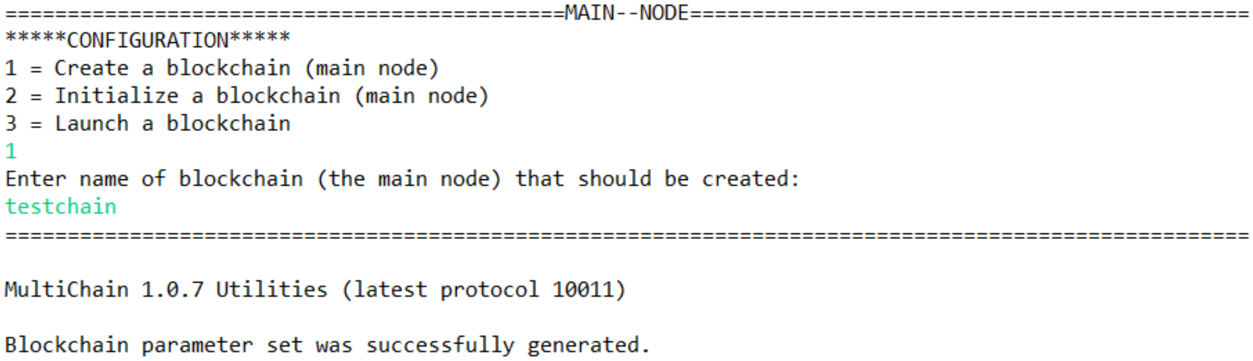
Configuration (only using a Main Node)[1]

1. Run “`MainNodeMain.java`”.



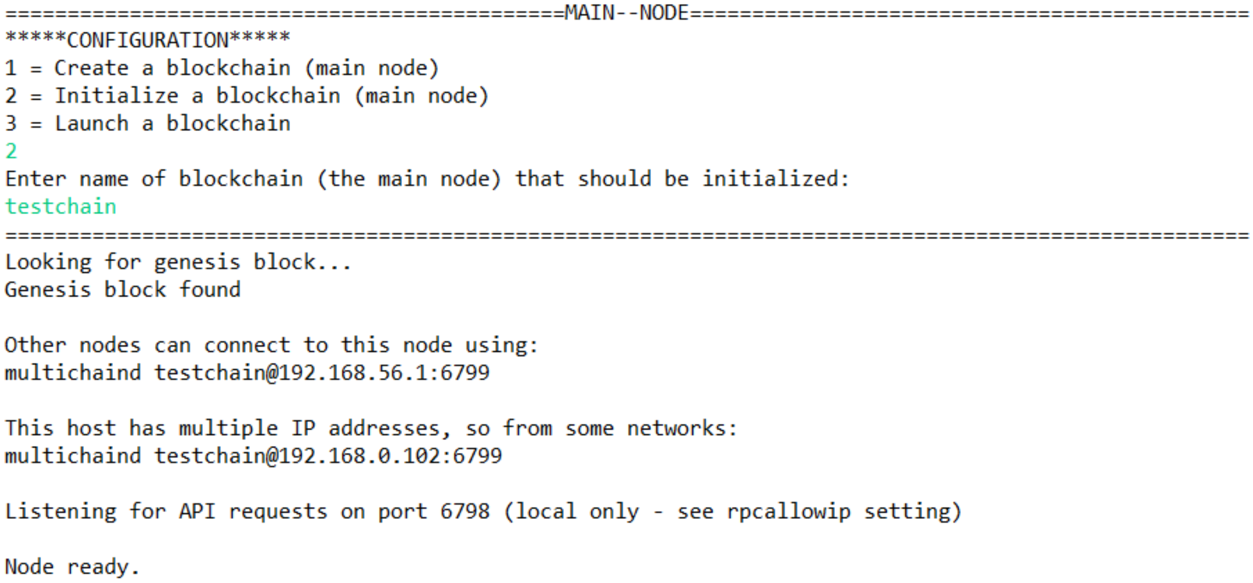
```
============================================MAIN--NODE============================================
*****CONFIGURATION*****
1 = Create a blockchain (main node)
2 = Initialize a blockchain (main node)
3 = Launch a blockchain
```

2. Create a main node by choosing the first option. This step is not needed when you already created a main node or when you are trying to access a node running on a different machine with a specific address.



```
============================================MAIN--NODE============================================
*****CONFIGURATION*****
1 = Create a blockchain (main node)
2 = Initialize a blockchain (main node)
3 = Launch a blockchain
1
Enter name of blockchain (the main node) that should be created:
testchain
==================================================================================================

MultiChain 1.0.7 Utilities (latest protocol 10011)

Blockchain parameter set was successfully generated.
```

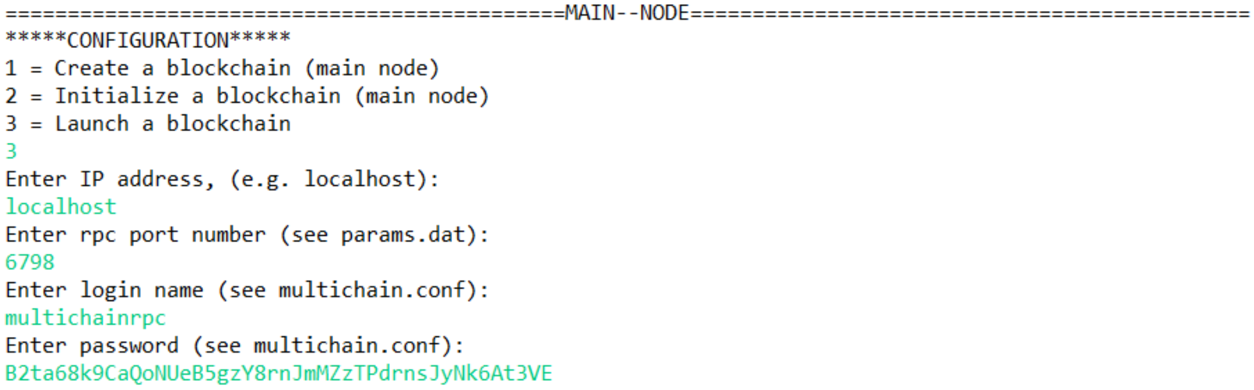
3. Initialize the main node by choosing the second option, there will be a daemon process running in the background (check using task manager). This step is not needed when the main node was already initialized or when you are trying to access a node running on a different machine with a specific address.



```
============================================MAIN--NODE============================================
*****CONFIGURATION*****
1 = Create a blockchain (main node)
2 = Initialize a blockchain (main node)
3 = Launch a blockchain
2
Enter name of blockchain (the main node) that should be initialized:
testchain
==================================================================================================
Looking for genesis block...
Genesis block found

Other nodes can connect to this node using:
multichaind testchain@192.168.56.1:6799

This host has multiple IP addresses, so from some networks:
multichaind testchain@192.168.0.102:6799

Listening for API requests on port 6798 (local only - see rpcallowip setting)

Node ready.
```

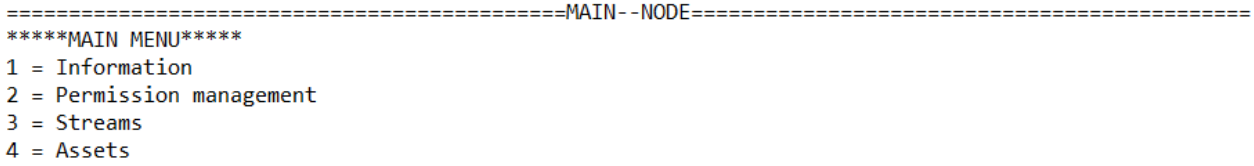
[1] The creation and initialization of nodes using the project currently works for Windows. On Mac or Linux, just run the commands using a terminal. See https://www.multichain.com/getting-started/ for help. Launching the blockchain and connecting to the API using the third option on either node should work on any operating system.

4. Now that the main node has been initialized, it is time to launch up the blockchain and connect with the API using the third option. The rpc port number can be found in the file "`params.dat`" and the login name and password in the file "`multichain.conf`" of the respective node (usually C:\Users\<User>\AppData\Roaming\MultiChain\<main_node>).



```
============================================MAIN--NODE============================================
*****CONFIGURATION*****
1 = Create a blockchain (main node)
2 = Initialize a blockchain (main node)
3 = Launch a blockchain
3
Enter IP address, (e.g. localhost):
localhost
Enter rpc port number (see params.dat):
6798
Enter login name (see multichain.conf):
multichainrpc
Enter password (see multichain.conf):
B2ta68k9CaQoNUeB5gzY8rnJmMZzTPdrnsJyNk6At3VE
```

5. You are now able to play around with the functionality of MultiChain provided by using the main node.

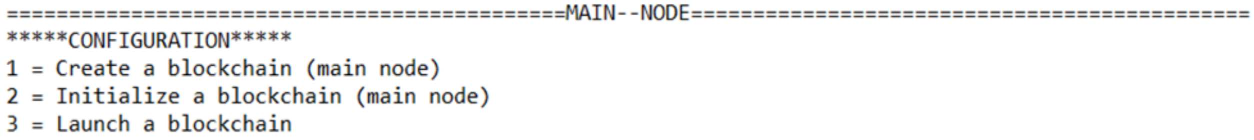


```
============================================MAIN--NODE============================================
*****MAIN MENU*****
1 = Information
2 = Permission management
3 = Streams
4 = Assets
```

Note: If you are trying to access a blockchain or node running on a different machine with a specific address, steps 2 and 3 can be omitted.

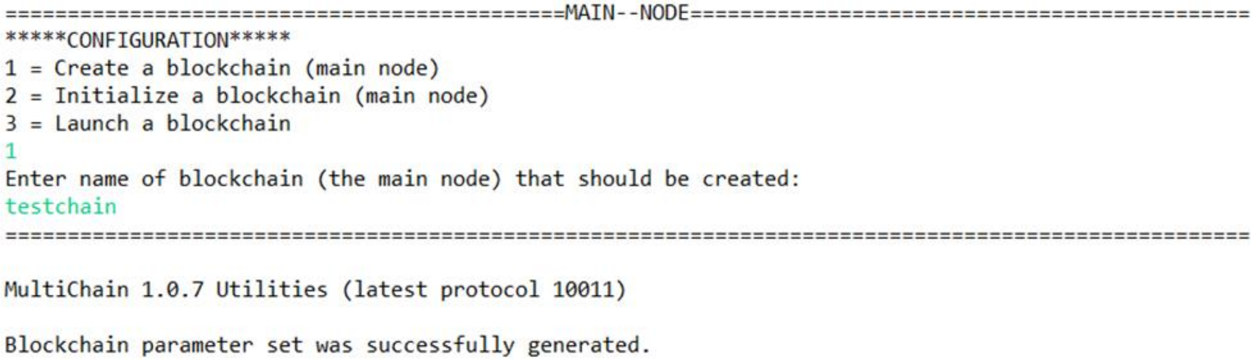
Advanced Configuration (using a Main node and a Secondary node)[1]

1. Run “`MainNodeMain.java`”.



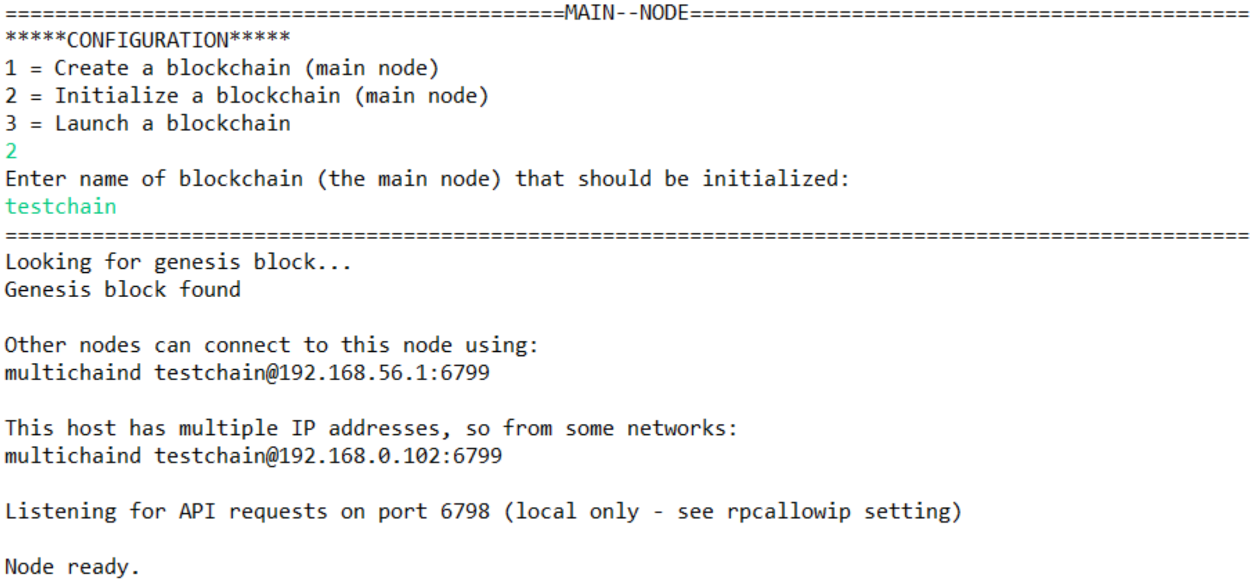
```
============================================MAIN--NODE============================================
*****CONFIGURATION*****
1 = Create a blockchain (main node)
2 = Initialize a blockchain (main node)
3 = Launch a blockchain
```

2. Create a main node by choosing the first option. This step is not needed when you already created a main node or when you are trying to access a node running on a different machine with a specific address.



```
============================================MAIN--NODE============================================
*****CONFIGURATION*****
1 = Create a blockchain (main node)
2 = Initialize a blockchain (main node)
3 = Launch a blockchain
1
Enter name of blockchain (the main node) that should be created:
testchain
==================================================================================================

MultiChain 1.0.7 Utilities (latest protocol 10011)

Blockchain parameter set was successfully generated.
```

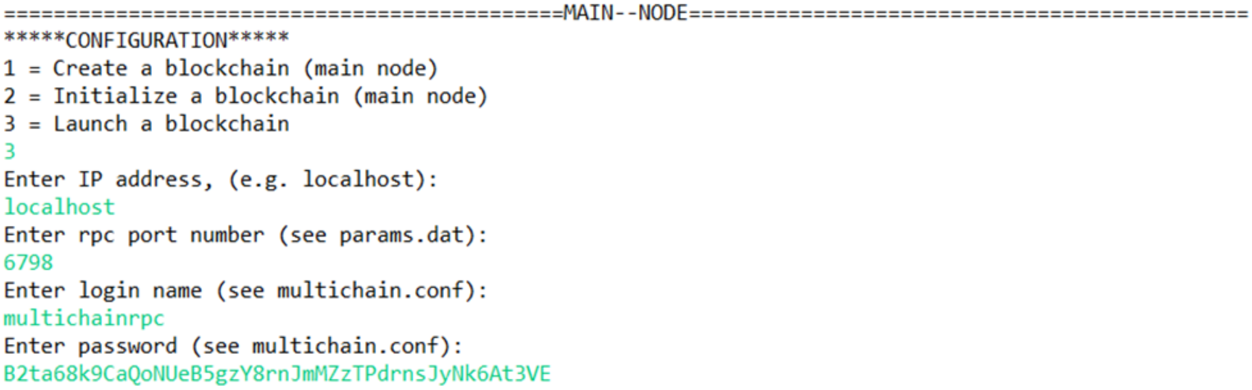
3. Initialize the main node by choosing the second option, there will be a daemon process running in the background (check using task manager). This step is not needed when the main node was already initialized or when you are trying to access a node running on a different machine with a specific address.



```
============================================MAIN--NODE============================================
*****CONFIGURATION*****
1 = Create a blockchain (main node)
2 = Initialize a blockchain (main node)
3 = Launch a blockchain
2
Enter name of blockchain (the main node) that should be initialized:
testchain
==================================================================================================
Looking for genesis block...
Genesis block found

Other nodes can connect to this node using:
multichaind testchain@192.168.56.1:6799

This host has multiple IP addresses, so from some networks:
multichaind testchain@192.168.0.102:6799

Listening for API requests on port 6798 (local only - see rpcallowip setting)

Node ready.
```

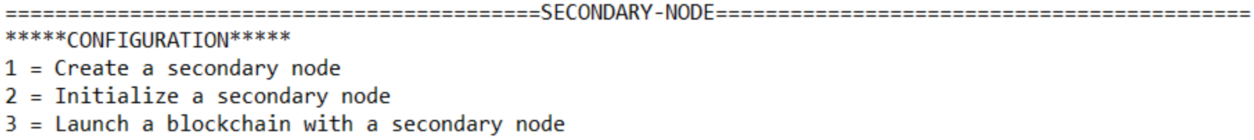
[1] The creation and initialization of nodes using the project currently works for Windows. On Mac or Linux, just run the commands using a terminal. See https://www.multichain.com/getting-started/ for help. Launching the blockchain and connecting to the API using the third option on either node should work on any operating system.

4. Now that the main node has been initialized, it is time to launch up the blockchain and connect with the API using the third option. The rpc port number can be found in the file "`params.dat`" and the login name and password in the file "`multichain.conf`" of the respective node (usually C:\Users\<User>\AppData\Roaming\MultiChain\<main_node>).



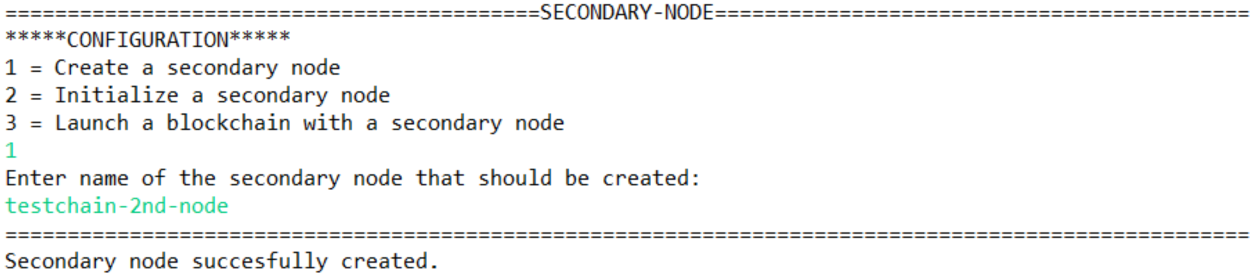
```
=============================================MAIN--NODE=============================================
*****CONFIGURATION*****
1 = Create a blockchain (main node)
2 = Initialize a blockchain (main node)
3 = Launch a blockchain
3
Enter IP address, (e.g. localhost):
localhost
Enter rpc port number (see params.dat):
6798
Enter login name (see multichain.conf):
multichainrpc
Enter password (see multichain.conf):
B2ta68k9CaQoNUeB5gzY8rnJmMZzTPdrnsJyNk6At3VE
```

5. Run "`SecondaryNodeMain.java`".



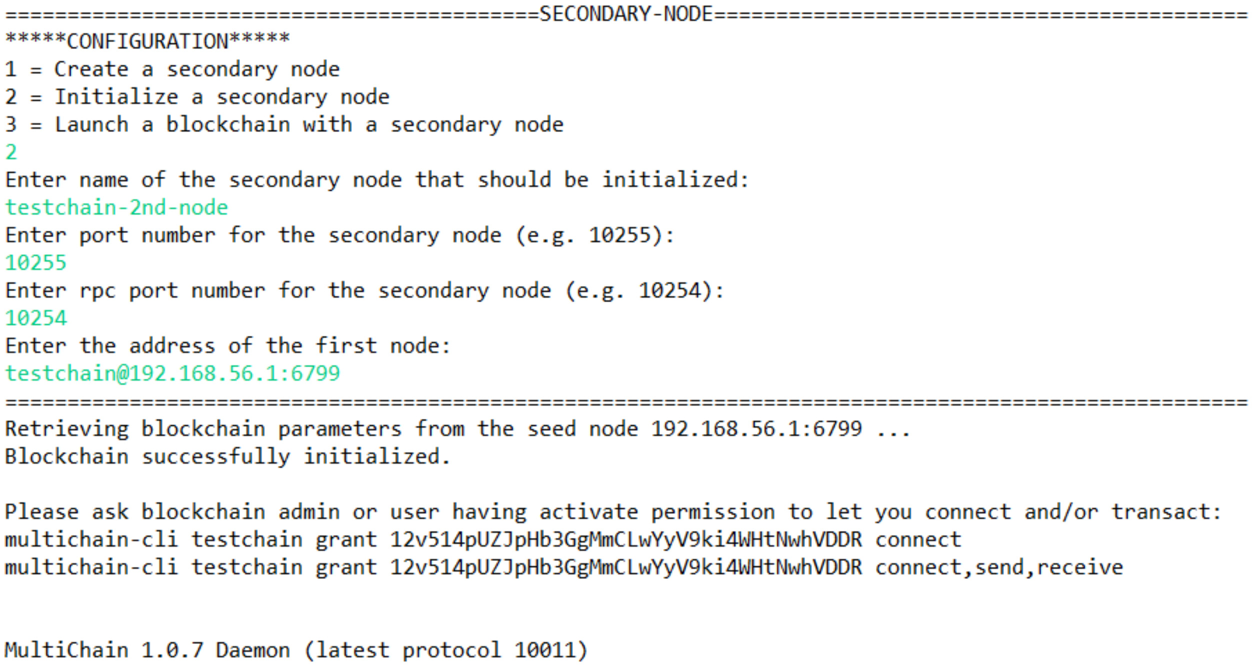
```
===========================================SECONDARY-NODE===========================================
*****CONFIGURATION*****
1 = Create a secondary node
2 = Initialize a secondary node
3 = Launch a blockchain with a secondary node
```

6. Create the secondary node by choosing the first option. This will just create a new directory under the specified path containing the chain data.



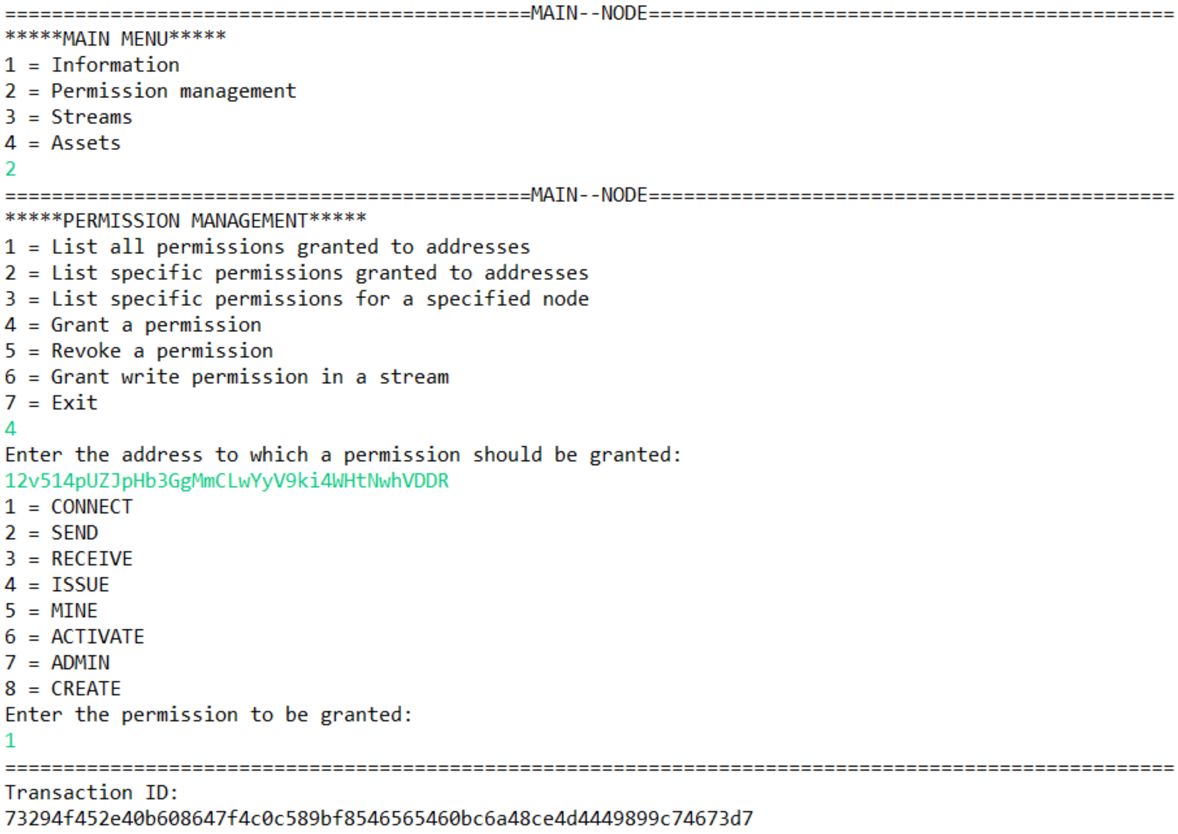
```
===========================================SECONDARY-NODE===========================================
*****CONFIGURATION*****
1 = Create a secondary node
2 = Initialize a secondary node
3 = Launch a blockchain with a secondary node
1
Enter name of the secondary node that should be created:
testchain-2nd-node
====================================================================================================
Secondary node succesfully created.
```

7. Initialize the secondary node by choosing the second option. The address of the first node can be found by calling the option "Information" and subsequently "General information about this blockchain" from the main menu of the main node, or at the time of initializing.



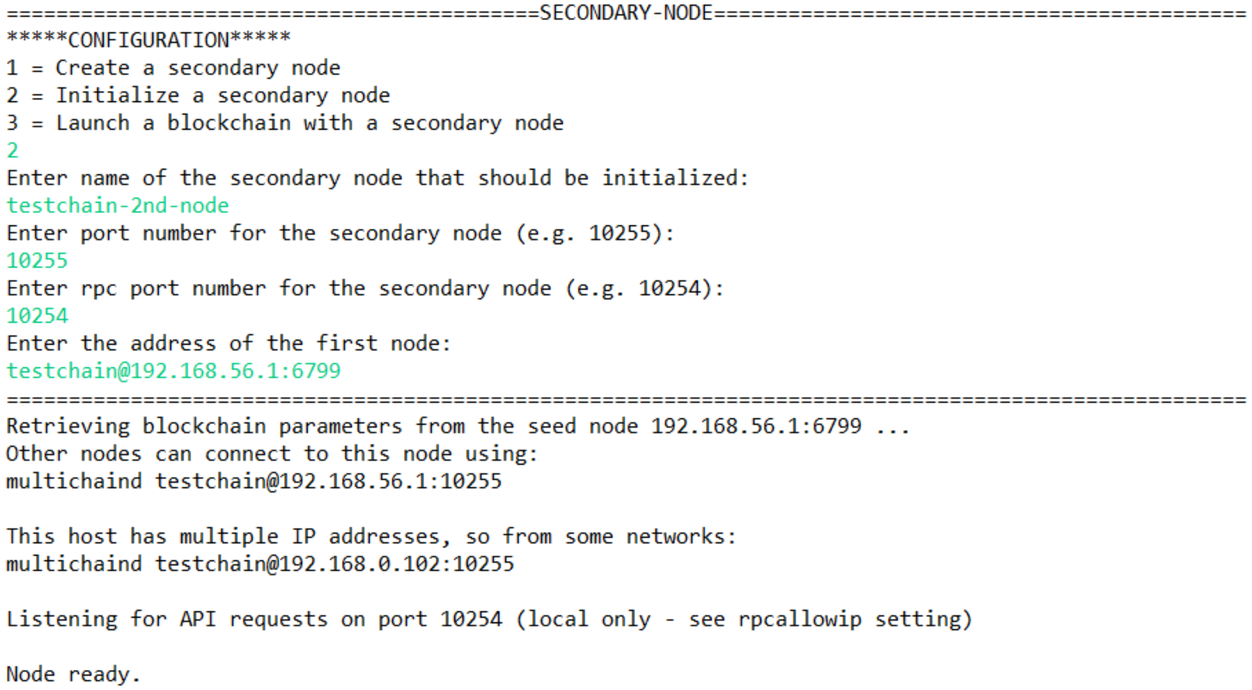
```
===========================================SECONDARY-NODE===========================================
*****CONFIGURATION*****
1 = Create a secondary node
2 = Initialize a secondary node
3 = Launch a blockchain with a secondary node
2
Enter name of the secondary node that should be initialized:
testchain-2nd-node
Enter port number for the secondary node (e.g. 10255):
10255
Enter rpc port number for the secondary node (e.g. 10254):
10254
Enter the address of the first node:
testchain@192.168.56.1:6799
====================================================================================================
Retrieving blockchain parameters from the seed node 192.168.56.1:6799 ...
Blockchain successfully initialized.

Please ask blockchain admin or user having activate permission to let you connect and/or transact:
multichain-cli testchain grant 12v514pUZJpHb3GgMmCLwYyV9ki4WHtNwhVDDR connect
multichain-cli testchain grant 12v514pUZJpHb3GgMmCLwYyV9ki4WHtNwhVDDR connect,send,receive


MultiChain 1.0.7 Daemon (latest protocol 10011)
```

8. Now you will have to grant connect permission from the main node to the secondary node. Select the option “Permission management” from the main menu of the main node, and then “Grant a permission”. Enter the address of the secondary node as displayed during initialization and choose the first option to grant connect permission.



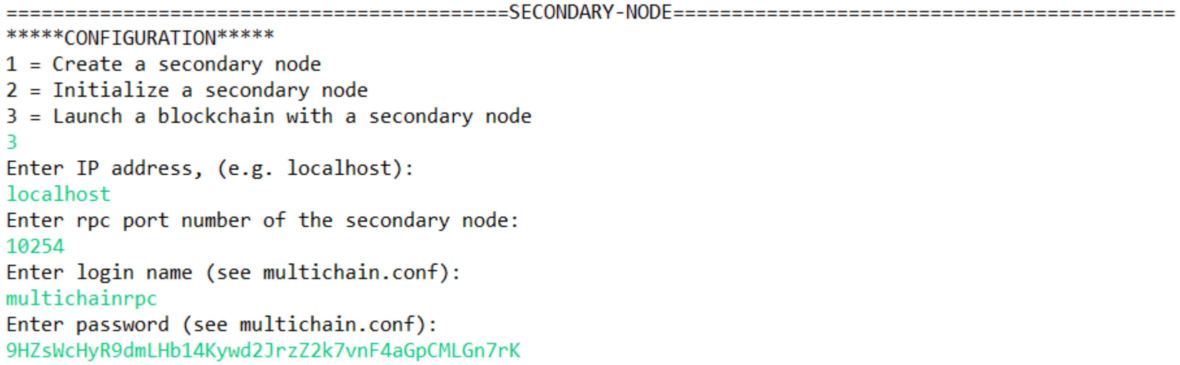
```
=============================================MAIN--NODE=============================================
*****MAIN MENU*****
1 = Information
2 = Permission management
3 = Streams
4 = Assets
2
=============================================MAIN--NODE=============================================
*****PERMISSION MANAGEMENT*****
1 = List all permissions granted to addresses
2 = List specific permissions granted to addresses
3 = List specific permissions for a specified node
4 = Grant a permission
5 = Revoke a permission
6 = Grant write permission in a stream
7 = Exit
4
Enter the address to which a permission should be granted:
12v514pUZJpHb3GgMmCLwYyV9ki4WHtNwhVDDR
1 = CONNECT
2 = SEND
3 = RECEIVE
4 = ISSUE
5 = MINE
6 = ACTIVATE
7 = ADMIN
8 = CREATE
Enter the permission to be granted:
1
====================================================================================================
Transaction ID:
73294f452e40b608647f4c0c589bf8546565460bc6a48ce4d4449899c74673d7
```

9. To connect the secondary node to the main node, return to the configuration menu of the secondary node and select the second option again (“Initialize a secondary node”). There will be a daemon process running in the background (check using task manager).



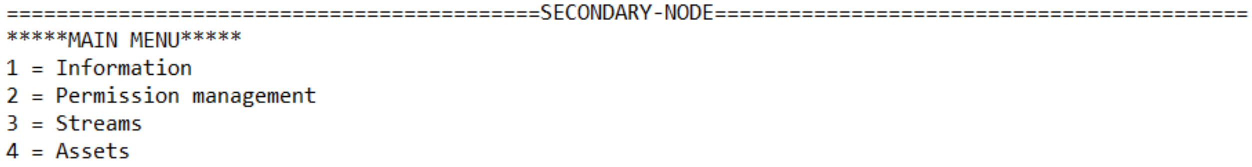
```
===========================================SECONDARY-NODE===========================================
*****CONFIGURATION*****
1 = Create a secondary node
2 = Initialize a secondary node
3 = Launch a blockchain with a secondary node
2
Enter name of the secondary node that should be initialized:
testchain-2nd-node
Enter port number for the secondary node (e.g. 10255):
10255
Enter rpc port number for the secondary node (e.g. 10254):
10254
Enter the address of the first node:
testchain@192.168.56.1:6799
====================================================================================================
Retrieving blockchain parameters from the seed node 192.168.56.1:6799 ...
Other nodes can connect to this node using:
multichaind testchain@192.168.56.1:10255

This host has multiple IP addresses, so from some networks:
multichaind testchain@192.168.0.102:10255

Listening for API requests on port 10254 (local only - see rpcallowip setting)

Node ready.
```

10. Now that the secondary node has connected to the main node, it is time to launch up the blockchain and connect with the API using the third option. The rpc port number is the same as specified during initialization and the login name and password can be found in the file “`multichain.conf`” of the respective node (usually C:\Users\<User>\AppData\Roaming\MultiChain\<secondary_node>\<main_node>).



```
===========================================SECONDARY-NODE===========================================
*****CONFIGURATION*****
1 = Create a secondary node
2 = Initialize a secondary node
3 = Launch a blockchain with a secondary node
3
Enter IP address, (e.g. localhost):
localhost
Enter rpc port number of the secondary node:
10254
Enter login name (see multichain.conf):
multichainrpc
Enter password (see multichain.conf):
9HZsWcHyR9dmLHb14Kywd2JrzZ2k7vnF4aGpCMLGn7rK
```

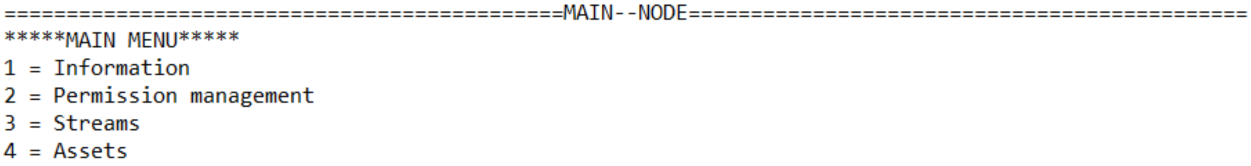
11. You are now able to play around with the functionality of MultiChain provided by using the main node and the secondary node.



```
===========================================SECONDARY-NODE===========================================
*****MAIN MENU*****
1 = Information
2 = Permission management
3 = Streams
4 = Assets
```

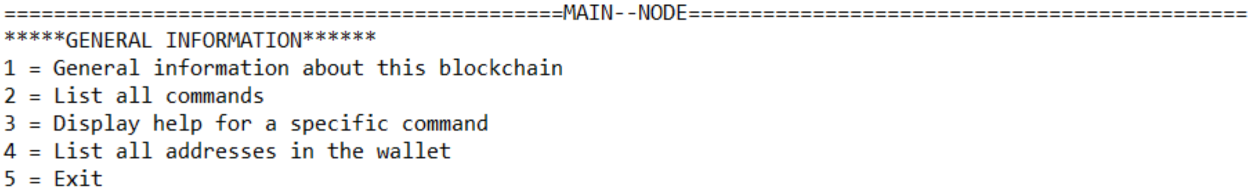
Menu Options for Simulation

Main Menu:



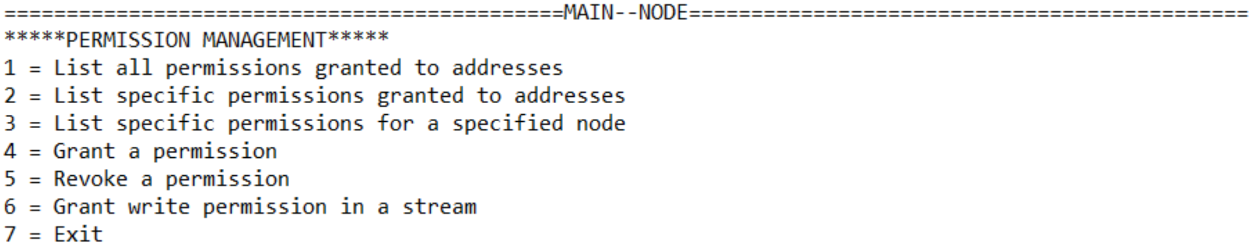
```
=============================================MAIN--NODE=============================================
*****MAIN MENU*****
1 = Information
2 = Permission management
3 = Streams
4 = Assets
```

Information:



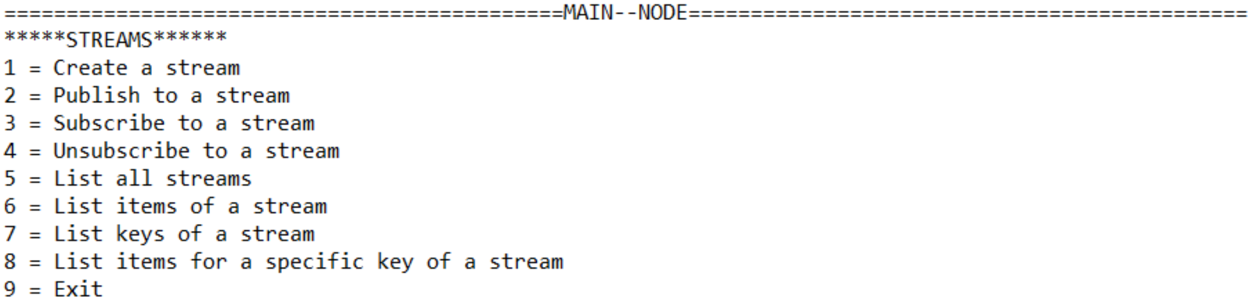
```
=============================================MAIN--NODE=============================================
*****GENERAL INFORMATION******
1 = General information about this blockchain
2 = List all commands
3 = Display help for a specific command
4 = List all addresses in the wallet
5 = Exit
```

Permission management:



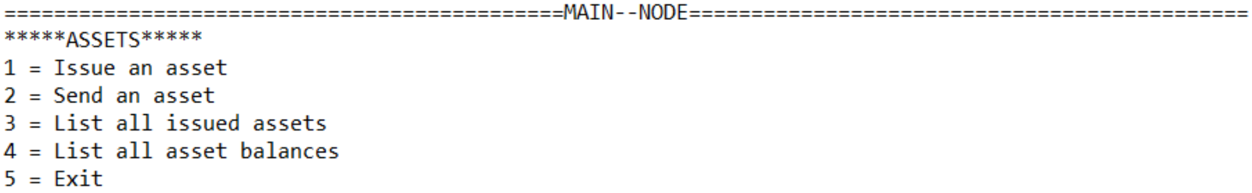
```
=============================================MAIN--NODE=============================================
*****PERMISSION MANAGEMENT*****
1 = List all permissions granted to addresses
2 = List specific permissions granted to addresses
3 = List specific permissions for a specified node
4 = Grant a permission
5 = Revoke a permission
6 = Grant write permission in a stream
7 = Exit
```

Streams:



```
=============================================MAIN--NODE=============================================
*****STREAMS******
1 = Create a stream
2 = Publish to a stream
3 = Subscribe to a stream
4 = Unsubscribe to a stream
5 = List all streams
6 = List items of a stream
7 = List keys of a stream
8 = List items for a specific key of a stream
9 = Exit
```

Assets:



```
=============================================MAIN--NODE=============================================
*****ASSETS*****
1 = Issue an asset
2 = Send an asset
3 = List all issued assets
4 = List all asset balances
5 = Exit
```

Visualization of Blockchain Activity using MultiChain Explorer

Installation (Windows)

1. Download MultiChain Explorer from https://github.com/MultiChain/multichain-explorer.
2. Unzip to desired folder.
3. Download and install Python version 2.x (version 3.x will not work) from https://www.python.org/downloads/.
4. Download and install PyCrypto for the corresponding version of Python from http://www.voidspace.org.uk/python/modules.shtml#pycrypto.
5. Download and install py-ubjson from https://pypi.org/project/py-ubjson/.
6. Install MultiChain Explorer by navigating to the directory containing the file "setup.py" and running it from the command line.

Installation (Linux)

1. Download MultiChain Explorer from https://github.com/MultiChain/multichain-explorer.
2. Install the following dependencies using:

```
sudo apt-get install sqlite3 libsqlite3-dev
sudo apt-get install python-dev
sudo apt-get install python-pip
sudo pip install --upgrade pip
sudo pip install pycrypto
sudo pip install py-ubjson
```

3. Install MultiChain Explorer by navigating to the directory containing the file "setup.py" and running:

```
python setup.py install --user
```

Configuration

1. Navigate to the file "params.dat" of the chain that was previously created and copy the value of "default-rpc-port".
2. Navigate to the file "multichain.conf" and add a line pointing to the rpc port to be used, e.g. "rpcport=1234".
3. Make a copy of the file "chain1.example.conf" in the folder of MultiChain Explorer and name it according to the blockchain. Configure the file as follows:
   - Change "port" to the port number for serving web pages (make sure your host's firewall allows traffic through that port).
   - Change "host" to 0.0.0.0 to serve web pages to anybody (make sure there is only a single host entry in the config file).
   - Change "dirname" to match the directory for your blockchain.
   - Change "chain" to set how the chain should be listed in the explorer.
   - Change "connect-args" for the location to store the explorer database.

Launching the Explorer

1. Load existing blockchain data into the explorer by navigating to the folder of MultiChain Explorer and running the following command (assuming the config file is named “chain1.conf”):

   ```
   python -m Mce.abe --config chain1.conf --commit-bytes 100000 --no-serve
   ```

   The output should be like:

   ```
   block_tx 1 1
   block_tx 2 2
   ...
   ```

2. Launch the explorer and serve web pages from your local computer by running the following command (assuming the config file is named “chain1.conf”):

   ```
   python -m Mce.abe --config chain1.conf
   ```

3. By default, the explorer will be listening for web requests on port 2750, unless you changed it in the Explorer's configuration file. In your browser visit http://localhost:2750/.