

Step1 - OpenST Utility Chain Sync

Now that you have all the tools, you will need the essential information to be able to connect and sync with the UC such as the following.

```
networkid=1409;
environment=sandbox;
assetPath="https://s3.amazonaws.com/assets.simpletoken.com/utility_chain
/$environment";
```

Proceed with the steps below.

1. Copy the following bash script and save it to a new file with the filename `setup_utility_chain_1409`

```
#!/bin/bash

networkid=1409;
environment=sandbox;

assetPath="https://s3.amazonaws.com/assets.simpletoken.com/utility_
chain/$environment";

GETH_EXEC=$(which geth);
if [[ -z $GETH_EXEC ]]; then
    echo "geth is not installed! You need to install geth to proceed
further."
    exit 1;
fi

read -p "Data dir path [$HOME]: " pathDir;

if [[ -z $pathDir ]]; then
    pathDir=$HOME;
fi

if [[ ! -d $pathDir ]]; then
    echo "Invalid path!!!";
    exit 1;
fi

datadir=$pathDir/uc_node_${networkid};

mkdir -p $datadir;

if [[ ! -d $datadir ]]; then
```

```
    echo "Error creating data dir!";
    exit 1;
fi

read -p "Network listening port (default: 30303) " port;
if [[ -z $port ]]; then
    port=30303;
fi

read -p "HTTP-RPC server listening port (default: 8545) " rpcport;
if [[ -z $rpcport ]]; then
    rpcport=8545;
fi

read -p "WS-RPC server listening port (default: 8546) " wsport;
if [[ -z $wsport ]]; then
    wsport=8546;
fi

read -p "Extra Args if any: " extraArgs;

# Download bootnode files
bn_filename="uc_${networkid}_boot_nodes.txt";
bn_src="$assetPath/$bn_filename";
bn_dest=$pathDir/$bn_filename;

wget -q $bn_src -O $bn_dest;
if [[ $? != 0 ]]; then
    echo "Unable to download bootnodes file!";
    exit 1;
fi

bootNodes=$(cat $bn_dest);
if [[ -z $bootNodes ]]; then
    echo "Invalid boot nodes!";
    exit 1;
fi

# rm -f $bn_dest;

# Download genesis file
filename="uc_${networkid}_genesis.json";
src_json="$assetPath/$filename";
dest_json=$pathDir/$filename;

wget -q $src_json -O $dest_json;
if [[ $? != 0 ]]; then
    echo "Genesis file download failed!";
    exit 1;
fi
```

```
$GETH_EXEC --datadir $datadir init $dest_json;
if [[ $? != 0 ]]; then
    echo "Not able to initialize genesis!";
    exit 1;
fi

# rm -f $dest_json;

$GETH_EXEC --networkid $networkid --datadir $datadir --port $port
--rpc --rpcapi eth,net,web3,personal --rpcport $rpcport --ws
```

```
--wsport $wsport --bootnodes $bootNodes $extraArgs;
```

The scripts in `setup_utility_chain_1409` do the following -

- Checks if `geth` is installed
- Creates a directory where the `chaindata` will be stored
- Connects `geth` to listen on the Network listening port (default: 30303)
- Connects `geth` to listen on the HTTP-RPC server listening port (default: 8545)
- Connects `geth` to listen on the WS-RPC server listening port (default: 8546)
- Downloads the Bootnode files
- Downloads the Genesis file
- Finally combines all the requirements together and initializes `geth` with the above parameters in the command below (already present in `setup_utility_chain_1409` - reexecution is not necessary).

```
$GETH_EXEC --networkid $networkid --datadir $datadir --port $port --rpc  
--rpcapi eth,net,web3,personal --rpcport $rpcport --ws --wsport $wsport  
--bootnodes $bootNodes $extraArgs;
```

Executing this script will sync your `geth` node and store the UC `chaindata` locally at `~/uc_node_1409/geth`. This might be different in your case depending on how you decide to set it up.

2. To execute the `setup_utility_chain_1409` bash script you must first modify the rights.

```
chmod 755 setup_utility_chain_1409
```

Here `chmod 755` is equal to `chmod u=rwx,go=rx` which means that the user can read, write and execute this script.

3. Execute the bash script by calling it and pressing `return` to accept all the bash script params, then allow your `geth` node to sync.

```
./setup_utility_chain_1409  
  
Data dir path [/Users/<user>]:  
Network listening port (default: 30303)  
HTTP-RPC server listening port (default: 8545)  
WS-RPC server listening port (default: 8546)  
Extra Args if any:
```

As the script starts the `geth` node sync with the UC, copy the inter-process communication (IPC) path displayed. Learn more about [inter-process communication](#). You will find the IPC endpoint path beside `IPC endpoint opened` in the console; copy the URL path, so that we can use it later for contract deployment. The URL path is the (local) path to attach a `geth` node from which you will later deploy sample contract on the UC.

The details below are specific to the user executing the script.

```
INFO [05-28|19:20:07] Maximum peer count
```

```
ETH=25
```

```
LES=0 total=25
INFO [05-28|19:20:07] Allocated cache and file handles
database=/Users/<user>/uc_node_1409/geth/chaindata cache=16 handles=16
INFO [05-28|19:20:08] Persisted trie from memory database      nodes=1
size=204.00B time=3.397µs gcnodes=0 gcsz=0.00B gctime=0s livenodes=1
livesize=0.00B
INFO [05-28|19:20:08] Successfully wrote genesis state
database=chaindata      hash=56b364...cffdd0
INFO [05-28|19:20:08] Allocated cache and file handles
database=/Users/noslav/uc_node_1409/geth/lightchaindata cache=16
handles=16
INFO [05-28|19:20:08] Persisted trie from memory database      nodes=1
size=204.00B time=4.137µs gcnodes=0 gcsz=0.00B gctime=0s livenodes=1
livesize=0.00B
INFO [05-28|19:20:08] Successfully wrote genesis state
database=lightchaindata
hash=56b364...cffdd0
INFO [05-28|19:20:08] Maximum peer count      ETH=25
LES=0 total=25
INFO [05-28|19:20:08] Starting peer-to-peer node
instance=Geth/v1.8.2-stable/darwin-amd64/go1.10
INFO [05-28|19:20:08] Allocated cache and file handles
database=/Users/noslav/uc_node_1409/geth/chaindata cache=768
handles=1024
INFO [05-28|19:20:08] Initialised chain configuration
config="{ChainID: 1409 Homestead: 1 DAO: <nil> DAOSupport: false EIP150:
2 EIP155: 3 EIP158: 3 Byzantium: 4 Constantinople: <nil> Engine:
clique}"
INFO [05-28|19:20:08] Initialising Ethereum protocol
versions="[63 62]" network=1409
INFO [05-28|19:20:08] Loaded most recent local header
number=173105 hash=5ab820...fa0ed5 td=346002
INFO [05-28|19:20:08] Loaded most recent local full block      number=0
hash=56b364...cffdd0 td=1
INFO [05-28|19:20:08] Loaded most recent local fast block
number=138286 hash=9c6c4e...d3f9a6 td=276364
INFO [05-28|19:20:08] Upgrading chain index
type=bloombits percentage=42
INFO [05-28|19:20:08] Loaded local transaction journal
transactions=0 dropped=0
INFO [05-28|19:20:08] Regenerated local transaction journal
transactions=0 accounts=0
INFO [05-28|19:20:08] Starting P2P networking
INFO [05-28|19:20:10] UDP listener up
self=enode://12c784bd9fc1a53b2fea73452de3c163ad83617a7b906f6bdc3fadee3a6
a70ef16be1346ebc7356906fb3977157abb1f4c7c1d36e6b7f2cec96f56cbf3603b49@[
:]:30303
INFO [05-28|19:20:10] RLPx listener up
self=enode://12c784bd9fc1a53b2fea73452de3c163ad83617a7b906f6bdc3fadee3a6
a70ef16be1346ebc7356906fb3977157abb1f4c7c1d36e6b7f2cec96f56cbf3603b49@[
```

```
:]:30303
```

```
INFO [05-28|19:20:10] IPC endpoint opened  
url=/Users/noslav/uc_node_1409/geth.ipc // ** <- COPY THIS PATH **  
  
INFO [05-28|19:20:10] HTTP endpoint opened  
url=http://127.0.0.1:8545 cors= vhosts=localhost  
INFO [05-28|19:20:10] WebSocket endpoint opened  
url=ws://127.0.0.1:8546
```

4. The sync may take several hours, depending on your system and your internet connection. To know if the sync is complete or not, open up a new console and pass in the following commands.

```
geth attach ~/uc_node_1409/geth.ipc // the IPC URL may be different in  
your case.  
  
> eth.syncing { startingBlock: 300, currentBlock: 312, highestBlock: 512  
}  
  
> eth.blockNumber 3628737
```

Take a look at [eth.syncing](#) and [eth.blockNumber](#) to know what they mean. The responses give the required information on the status of node's sync with the UC. Make sure you are in sync with the latest blocks before proceeding with deploying contracts on the UC.