HyperCash Omni Protocol Public Test Operation Manual

◆ Environmental construction

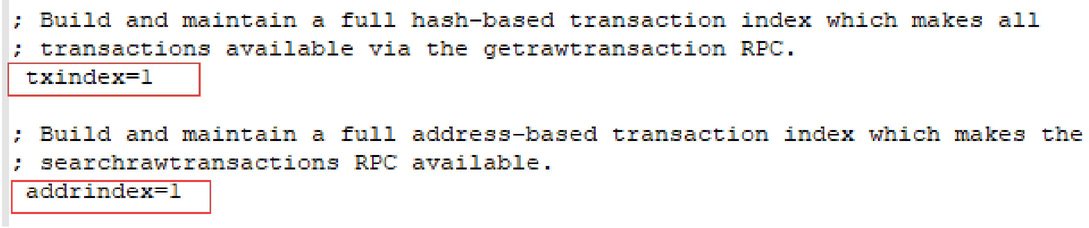
1. Download **hcd**, **hcwallet**, and **hcOMNI** by means of:

   a) Binary files
   https://github.com/HcashOrg/hcOMNI/releases/tag/RC1.0

   OR

   b) Source code
   hcd: https://github.com/HcashOrg/hcd/tree/omnibranch
   hcwallet: https://github.com/HcashOrg/hcwallet/tree/omnibranch
   hcOMNI: https://github.com/HcashOrg/hcOMNI

2. Modify the **hcd** configuration file, enter **"%localappdata%"** in the command line, open the **Hcd** folder (if there is no hcd folder, you will need to first start hcd (after which a folder will generate automatically), then open the **hcd.conf** file, find **"addrindex=1"** and **"txindex=1"** , and remove the **";"** in front of both **"addrindex=1"** and **"txindex=1"**. See below:



```
; Build and maintain a full hash-based transaction index which makes all
; transactions available via the getrawtransaction RPC.
 txindex=1

; Build and maintain a full address-based transaction index which makes the
; searchrawtransactions RPC available.
 addrindex=1
```

3. Start **hcd** with the following command:

   Hcd --testnet -u admin -P 123

4. Start **hcwallet** with the following command;

   hcwallet --testnet -u admin -P 123 --hcusername admin --walletpass 123 --hcpassword 123

5. Create a new wallet address using the following command:

   hcctl --testnet -u admin -P 123 --wallet getnewaddress

6. Open https://faucet.h.cash/requestfaucet, paste your newly generated address (from step 5) into the fillable bar and click the ‘send’ button to receive 2 HC. ***Note*** This is the recommended way to continue receiving test HC.

◆ Create a fixed supply of tokens

1. Issue 1,000,000 tokens with the following command：

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_sendissuancefixed
   "TsSmoC9HdBhDhq4ut4TqJY7SBjPqJFAPkGK" 2 1 0 "Companies""Bitcoin
   Mining""Quantum Miner""""""1000000"
   ```

2. Use the following command to query whether the issuance was successful. If a new block is generated, this confirms the issuance.

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_listproperties
   ```

3. If the issuance is successful, a corresponding **propertyid** will be generated (**propertyid** is the token property ID, which is used to distinguish different types of tokens). From here you can transfer 1000 Omni tokens to two specified addresses through the following command:

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_send
   TsSmoC9HdBhDhq4ut4TqJY7SBjPqJFAPkGK
   Tsbi4Wq9xG7YHwnqWGV17fFDYBrPvmhr7hg  2147483651 1000
   ```

4. You can then query the balance of the two addresses through the following command. ***Note*** Like above, this also requires a new block.

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_getbalance
   TsSmoC9HdBhDhq4ut4TqJY7SBjPqJFAPkGK 2147483651
   ```

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_getbalance
   Tsbi4Wq9xG7YHwnqWGV17fFDYBrPvmhr7hg 2147483651
   ```

◆ Create a manageable supply of tokens

Manage the token supply with the following commands:

1. Create a **propertyid: *Note*** ensure the address belongs to the token manager.

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_sendissuancemanaged
   "TsSmoC9HdBhDhq4ut4TqJY7SBjPqJFAPkGK"2 1 0 "Companies1""Bitcoin
   Mining1""Quantum Miner1"""""
   ```

2. Similar to the above section, use the following command to check whether the **propertyid** has been generated:

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_listproperties
   ```

3. View **propertyid** details with the following command:

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_getproperty 2147483817
   ```

4. 1000 Omni tokens can be issued to a specified address through the following command: *Note* The former stated address is the managers, and the latter is the address of the tokens.

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_sendgrant
   "TsSmoC9HdBhDhq4ut4TqJY7SBjPqJFAPkGK"
   "Tsbi4Wq9xG7YHwnqWGV17fFDYBrPvmhr7hg" 2147483817 1000
   ```

5. View the details of the token issuance through the following command:

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_getgrants 2147483817
   ```

6. Check the balance of the specified address (as stated in step 4) through the following command:

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_getbalance
   Tsbi4Wq9xG7YHwnqWGV17fFDYBrPvmhr7hg 2147483817
   ```

7. Refer to "**Create a fixed supply of tokens**" (Step 3) for details regarding token trading.

◆ How to withdraw a manageable supply of tokens

1. Users can withdraw 1000 manageable supply tokens through the following command: *Note* Only tokens from the managers address can be recalled.

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_sendrevoke
   TsSmoC9HdBhDhq4ut4TqJY7SBjPqJFAPkGK 2147483817 1000
   ```

2. View withdraw details through the following command:

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_getgrants 2147483817
   ```

◆ How to Create Crowdfunded Tokens (i.e. HC and Omni tokens swap)

1. Users can obtain Omni tokens automatically to their own address by sending HC tokens to the manager address through the following command:

   ```
   Hcctl --testnet -u admin -P 123 --wallet sendfromaddresstoaddress
   TsY5aGpApd5snxXm2yVMQ7AsXHzoK2kX1fo TsSmoC9HdBhDhq4ut4TqJY7SBjPqJFAPkGK
   0.2
   ```

2. Find the exchanged (swapped) Omni tokens through the following command:

   ```
   Hcctl --testnet -u admin -P 123 --wallet omni_getbalance
   TsY5aGpApd5snxXm2yVMQ7AsXHzoK2kX1fo 2
   ```

3. Sell Omni tokens through the following command: *Note* The order will only be finalized when a buy order comes in.

```
Hcctl --testnet -u admin -P 123 --wallet omni_senddexsell
TsY5aGpApd5snxXm2yVMQ7AsXHzoK2kX1fo 2 100 1 255 0.0002 1
```

4. Other users can buy Omni tokens through the following command:

```
Hcctl --testnet -u admin -P 123 --wallet omni_senddexaccept
TsiuTUYVEijspqRNqpKe9uf243k1AuAVjjE TsY5aGpApd5snxXm2yVMQ7AsXHzoK2kX1fo 2
20
```

```
Hcctl --testnet -u admin -P 123 --wallet sendfromaddresstoaddress
TsiuTUYVEijspqRNqpKe9uf243k1AuAVjjE TsY5aGpApd5snxXm2yVMQ7AsXHzoK2kX1fo
0.2
```

5. Use the following command to query Omni token buy/sell details:

```
Hcctl --testnet -u admin -P 123 --wallet omni_getbalance
TsY5aGpApd5snxXm2yVMQ7AsXHzoK2kX1fo 2
```

◆ How to create a crowdfunded token (A-type Omni token swapped with B-type Omni token)

1. Start a crowdfunding transaction through the following command:

```
Hcctl --testnet -u admin -P 123 --wallet omni_sendissuancecrowdsale "
TsSmoC9HdBhDhq4ut4TqJY7SBjPqJFAPkGK" 1 1 0 "Comp" "Btc" "Quantum" "" "" 1
"100" 1546275661 30 15
```

2. Send the A-type token to the address in the previous step through the following command: *Note* When the transaction is completed, the user will receive B-type Omni tokens in return.

```
Hcctl --testnet -u admin -P 123 --wallet omni_send
TsY5aGpApd5snxXm2yVMQ7AsXHzoK2kX1fo TsSmoC9HdBhDhq4ut4TqJY7SBjPqJFAPkGK
5 1000
```

3. Find the Omni token in the address by the following command

```
Hcctl --testnet -u admin -P 123 --wallet omni_getbalance
TsiuTUYVEijspqRNqpKe9uf243k1AuAVjjE 2
```

◆ Main API description

◆ omni_send

Create and broadcast a simple send transaction. Arguments as below:

Name	Type	Presence	Description
fromaddress	string	Required	Address to send
toaddress	string	Required	Address to receive
propertyid	number	Required	The property identifier
amount	string	Required	Amount to send
redeemaddress	string	Optional	
referenceamount	string	Optional	

◆ omni_getbalance

Returns the token balance to a specified address and propertyid. Arguments as below:

Name	Type	Presence	Description
address	string	Required	The address
propertyid	number	Required	The property identifier

◆ omni_sendissuancefixed

Creates a new fixed supply token. Arguments as below:

Name	Type	Presence	Description
fromaddress	string	Required	Address to send
ecosystem	number	Required	The ecosystem to create the tokens in (1 for main ecosystem, 2 for test ecosystem)
type	number	Required	The type of the tokens to create: (1 for indivisible tokens, 2 for divisible tokens)
previousid	number	Required	An identifier of a predecessor token (0 for new tokens)
category	string	Required	A category for the new tokens (can be "")
subcategory	string	Required	A subcategory for the new tokens (can be "")
name	string	Required	The name of the new tokens to create
url	string	Required	An URL for further information about the new tokens (can be "")
data	string	Required	A description for the new tokens (can be "")
amount	string	Required	The number of tokens to be created
tokensperunit	string	Required	Number of tokens awarded by units invested in crowdfunding
deadline	number	Required	Crowdfunding deadline as a Unix timestamp
earlybonus	number	Required	Provide participants with weekly percentages of early bonuses.
issuerpercentage	number	Required	Percentage of tokens to be awarded to the issuer

◆ omni_sendissuancemanaged

Creates a manageable supply of tokens. Arguments as below:

Name	Type	Presence	Description
fromaddress	string	Required	Address to send
ecosystem	number	Required	The ecosystem to create the tokens in (1 for main ecosystem, 2 for test ecosystem)
type	number	Required	The type of the tokens to create: (1 for indivisible tokens, 2 for divisible tokens)
previousid	number	Required	An identifier of a predecessor token (0 for new tokens)
category	string	Required	A category for the new tokens (can be "")
subcategory	string	Required	A subcategory for the new tokens (can be "")
name	string	Required	The name of the new tokens to create
url	string	Required	An URL for further information about the new tokens (can be "")
data	string	Required	A description for the new tokens (can be "")

◆ omni_sendissuancecrowdsale

Creates crowdfunded tokens. Arguments as below:

Name	Type	Presence	Description
fromaddress	string	Required	Address to send
ecosystem	number	Required	The ecosystem to create the tokens in (1 for main ecosystem, 2 for test ecosystem)
type	number	Required	The type of the tokens to create: (1 for indivisible tokens, 2 for divisible tokens)
previousid	number	Required	An identifier of a predecessor token (0 for new tokens)
category	string	Required	A category for the new tokens (can be "")
subcategory	string	Required	A subcategory for the new tokens (can be "")
name	string	Required	The name of the new tokens to create
url	string	Required	An URL for further information about the new tokens (can be "")
data	string	Required	A description for the new tokens (can be "")
propertyiddesired	number	Required	Identifier of tokens eligible to participate in crowdfunding
tokensperunit	string	Required	Number of tokens awarded by units invested in crowdfunding
deadline	number	Required	Crowdfunding deadline as a Unix timestamp
earlybonus	number	Required	Provide participants with weekly percentages of early bonuses
issuerpercentage	number	Required	Percentage of tokens to be awarded to the issuer