Factom Common Tasks

This documentation is assuming use of Factom Java API. The steps can be ported to other implementations, but the syntax will be the Factom Java API

Send Factoid

What you will need

A temporary transaction name. [tranName]

You pick it. It will only exist as long as you are building the transaction.

Address to send from. [fromAddr]

You can use the address or name you gave in your wallet (assuming you are using
a local fctwallet)

Address to send to. [toAddr]

If this is your local (fctwallet) address, you can use the name or address. If this is a
remote address, we are not going to be able to translate the name to an
address. USE THE ADDRESS.

Amount to send. [amount]

This is in factoshi. similar to satoshi. 1 factoshi = 1/100000000 factoid. You will
also need a balance in the sending address.

Steps

1. Delete Transaction
   ```
   DeleteTransaction(tranName);
   ```
2. Create Transaction
   ```
   NewTransaction(tranName);
   ```
3. AddInput
   ```
   AddInput(tranName,fromAddr,amount);
   ```
4. AddOutput
   ```
   AddOutput(tranName,toAddr,amount);
   ```
5. GetFee - mostly informational
   ```
   GetFee(tranName);
   ```
6. AddFee
   ```
   AddFee(tranName,fromAddr);
   ```
7. Sign Transaction
   ```
   SignTransaction(tranName);
   ```
8. Submit Transaction
   ```
   SubmitTransaction(tranName);
   ```

If you are only going to have one output and one input, just use:

```
SendFactoidsFullTransaction(fromAddr,toAddr,FactoidAmount);
```

If you have more than one input or output, you have to use the multi-step method.

Buy Entry Credits

This is very similar to Send Factoid.

What you will need

A temporary transaction name. [tranName]

You pick it. It will only exist as long as you are building the transaction.

Address to send from. [fromAddr]

You can use the address or name you gave in your wallet (assuming you are using a local fctwallet) This is a FACTOID ADDRESS

Address to send to. [toAddr]

If this is your local (fctwallet) address, you can use the name or address. If this is a remote address, we are not going to be able to translate the name to an address. USE THE ADDRESS. THIS IS AN ENTRY CREDIT ADDRESS

Amount to send. [amount]

This is in factoshi. similar to satoshi. 1 factoshi = 1/100000000 factoid. You will also need a balance in the sending address. We only speak factoshi. If you send 1 factoid (100,000,000 factoshi), you will receive 1 factoid worth of entry credits according to the current exchange rate. (`GetExchangeRate()`) If you need a specific number of Entry Credits, call this first and calculate how many factoids are needed.

Steps

1. Delete Transaction - just in case there is a partial transaction under that name

```
DeleteTransaction(tranName);
```

2. Create Transaction

```
NewTransaction(tranName);
```

3. Add Input

```
AddInput(tranName,fromAddr,amount);
```

4. Add EC Output

```
AddECOutput(tranName,toAddr,amount);
```

5. Get Fee - mostly informational

```
GetFee(tranName);
```

6. Add Fee

```
AddFee(tranName,fromAddr);
```

7. Sign Transaction

```
SignTransaction(tranName);
```

8 Submit Transaction

```
SubmitTransaction(tranName);
```

If you are only going to have one output and one input, just use: (amount is factoid)

```
      BuyEntryCreditsFullTransaction(fromAddr,toAddr,FactoidAmount);
```

If you have more than one input or output, you have to use the multi-step method.

Make a new Chain.

Chains require a first entry. The difference between creating a Chain and creating an Entry is that the Entry you are creating is either being attached to an existing Chain or creating a new Chain.

```
ComposeChainCommit(EntryCreditAddress, ExternalIDs , EntryBody);
```

Entry Credit Address is a String. Use the Address, not the Wallet Label.

ExternalIDs is an Array of Strings. (these will be used to create chain id)

Entry Body is a String. It is freeform data to be added to this entry.

This will be creating the first Entry into the Chain also.

Should return the Chain ID for use in future entries.

Make a new Entry.

This requires an existing Chain.

```
ComposeChainCommit(EntryCreditAddress, ChsinID, ExternalIDs ,
EntryBody);
```

Entry Credit Address is a String. Use the Address, not the Wallet Label.

ChainID is a string. 64 character hex.

ExternalIDs is an Array of Strings. (these will be used to create chain id)

Entry Body is a String. It is freeform data to be added to this entry.

returns 200 OK status on success.