Pyrk Simple Tokens: Specification

Michael Osullivan mike@pyrk.org

This document describes how Pyrk Simple Tokens will be implemented on the Pyrk blockchain. Pyrk tokens are user created tokens. The supply of a token is completely controlled by the end user and no interaction with Pyrk core mining other than confirming the blocks that these transactions are contained in.

Pyrk Core has an available OP_RETURN size of 160 bytes (HEX).

We will employ a Tokennode network to assist in maintaining consensus for Tokens. Users may use their own Tokennodes if they do not wish to use the publicly available nodes.

This specification is "loosely" based on the Colored Coins protocol.

OP_CODES

OP_CODE	Type	Description	Comments
"0x01"	GENESIS	Issue Token (Genesis)	Create new token
"0x02"	ADDMETA	Add Metadata	Adds information about a token
"0x03"	BURN	Burn Token	Remove value from circulation
"0x04"	SEND	Send Token	Transfer tokens to another address
"0x05"	PAUSE	Pause Token	Prevent new transactions
"0x06"	RESUME	Resume Token	Allow new transactions after pause
"0x07"	NEWOWNER	Transfer Token Ownership	Assigns token to new owner address
"0x08"	AUTHMETA	Allow address to add meta	
"0x09"	REVOKEMETA	Revoke meta add access	

Byte Positions

In each OP_RETURN transaction, it is specified the number of bytes allowed. If your data is less than Max Bytes, then Pad-Left your data with null character. The final item does not need to be padded.

Issue Token (Genesis)

Bytes	Description	Conversion Type	Comments
2	Protocol ID	N/A	0x3432 (aka “42”)
1	Version Number	N/A	Currently 0x01 (aka version “1”)
1	OP_CODE for issuance	N/A	Op code 0x01
5	Ticker Code.	UTF-8 to HEX	Should be 1 to 5 Chars
20	Token Name	UTF-8 to HEX	
8	Issue Value in Satoshis	INT to HEX	Max allowed value 184 billion
35	Document URI	UTF-8 to HEX	example: https://www.pyrk.org
80	Logo URI	UTF-8 to HEX	example https://www.pyrk.org/logo.png

To create the token, a 0 amount transaction is created (plus fee) and sent to the address which you want to be the owner of the token. Upon transaction confirmation and receipt of the issuance, your wallet client will also tell you the “Token ID”, which is an identifier on the Pyrk network for that token.

The token id is calculated the following way and comprises of 22 hexadecimal bytes:

First 4 bytes of owner address after converting to hex + last 9 bytes of block hash the token transaction was included in + first 9 bytes of the transaction id the token was created with.

If you are wanting to make a non-fungible token for meta data use only, then set the issue value to 0.

Send Token

Bytes	Description	Conversion Type	Comments
2	Protocol ID	HEX	0x3432 (aka “42”)
1	Version Number	HEX	Currently 0x01 (aka version “1”)
1	OP_CODE for Send	HEX	Op code 0x04
22	Token ID	HEX	Token Hex ID
8	Send Value in Satoshis	INT to HEX	Amount to send. Max 184 Billion
34	Recipient Address	UTF-8 to HEX	Receivers Pyrk token address
20	Payment ID	UTF-8 to HEX	optional

A send transaction contains 1 input and 2 outputs.

Input 1: This input must be from the address which holds the token you are sending, it must also have enough Pyrk to pay the transaction fee
Output 1: This is the address receiving the change transaction. Same as input.
Output 2: This is the address with the OP_RETURN data about the token transfer.

Transaction values are validated prior to broadcasting to the blockchain using Tokennode Quorum. If an invalid transaction is broadcast to the network, the sender will pay the transaction fee, but the balances will not update, as the validation in the token system will fail even though the top layout transaction itself is accepted.

Burn Token

Bytes	Description	Conversion Type	Comments
2	Protocol ID	N/A	0x3432 (aka “42”)
1	Version Number	N/A	Currently 0x01 (aka version “1”)
1	OP_CODE for Burn	N/A	Op code 0x03
22	Token ID	N/A	Token Hex ID
8	Burn Value in Satoshis	INT to HEX	Max value 184 Billion

A burn transaction contains 1 input and 2 outputs.

Input 1: This input must be from the address which holds the token you are sending, it must also have enough Pyrk to pay the transaction fee
Output 1: This is the address receiving the change transaction. Same as input.
Output 2: This is the address with the OP_RETURN data about the token burn.

Add Metadata

Bytes	Description	Conversion Type	Comments
2	Protocol ID	N/A	0x3432 (aka “42”)
1	Version Number	N/A	Currently 0x01 (aka version “1”)
1	OP_CODE for Metadata	N/A	Op code 0x02
4	META_CODE	N/A	ie, 0x00000001 is update Document URI
22	Token ID	N/A	Token Hex ID
130	Value of metadata	UTF-8 to HEX	Maximum 130 bytes of metadata

An add metadata transaction contains 1 input and 2 outputs.

Input 1: This input must be from the address which holds the token you are sending, it must also have enough Pyrk to pay the transaction fee
Output 1: This is the address receiving the change transaction. Same as input.
Output 2: This is the address with the OP_RETURN data about the token burn.

Metadata META_CODES

META_CODE	Description	Comments
“0x00000001”	Document URI	example, a website url
“0x00000002”	Logo URI	URI to a logo file. 50x50 px PNG only
“0x00000003” - 0xFFFFFFFF	General Meta	For general use, you define what they mean

Pause Token

Bytes	Description	Comments
2	Protocol ID	0x3432 (aka “42”)
1	Version Number	Currently 0x01 (aka version “1”)
1	OP_CODE for Pause	Op code 0x05
22	Token ID	Token Hex ID

An add metadata transaction contains 1 input and 2 outputs.

Input 1: This input must be from the address which holds the token you are sending, it must also have enough Pyrk to pay the transaction fee
Output 1: This is the address receiving the change transaction. Same as input.
Output 2: This is the address with the OP_RETURN data about the token burn.

Resume Token

Bytes	Description	Comments
2	Protocol ID	0x3432 (aka “42”)
1	Version Number	Currently 0x01 (aka version “1”)
1	OP_CODE for Resume	Op code 0x06
22	Token ID	Token Hex ID

An add metadata transaction contains 1 input and 2 outputs.

Input 1: This input must be from the address which holds the token you are sending, it must also have enough Pyrk to pay the transaction fee
Output 1: This is the address receiving the change transaction. Same as input.
Output 2: This is the address with the OP_RETURN data about the token burn.

New Ownership

Bytes	Description	Conversion Type	Comments
2	Protocol ID	N/A	0x3432 (aka “42”)
1	Version Number	N/A	Currently 0x01 (aka version “1”)
1	OP_CODE for New Owner	N/A	Op code 0x07
22	Token ID	N/A	Token Hex ID
34	New Owner Address	UTF-8 to HEX	Pyrk Address

An add metadata transaction contains 1 input and 2 outputs.

Input 1: This input must be from the address which holds the token you are sending, it must also have enough Pyrk to pay the transaction fee
Output 1: This is the address receiving the change transaction. Same as input.
Output 2: This is the address with the OP_RETURN data about the token burn.

Authorize Meta

Bytes	Description	Conversion Type	Comments
2	Protocol ID	N/A	0x3432 (aka “42”)
1	Version Number	N/A	Currently 0x01 (aka version “1”)
1	OP_CODE for Authmeta	N/A	Op code 0x08
22	Token ID	N/A	Token Hex ID
34	Address giving authorization to	UTF-8 to HEX	

An add metadata transaction contains 1 input and 2 outputs.

Input 1: This input must be from the address which holds the token you are sending, it must also have enough Pyrk to pay the transaction fee
Output 1: This is the address receiving the change transaction. Same as input.
Output 2: This is the address with the OP_RETURN data about the token burn.

Revoke Meta

Bytes	Description	Conversion Type	Comments
2	Protocol ID	N/A	0x3432 (aka “42”)
1	Version Number	N/A	Currently 0x01 (aka version “1”)
1	OP_CODE for Revokemeta	N/A	Op code 0x09
22	Token ID	N/A	Token Hex ID
34	Address revoking access to meta	UTF-8 to HEX	

An add metadata transaction contains 1 input and 2 outputs.

Input 1: This input must be from the address which holds the token you are sending, it must also have enough Pyrk to pay the transaction fee
Output 1: This is the address receiving the change transaction. Same as input.
Output 2: This is the address with the OP_RETURN data about the token burn.

FEES

The fee to Issue a new token will initially be set at 5 PYRK. For all other actions, standard transaction network fees apply. Fees generated from new token issuance are given to the miners in newly mined blocks which contain the new issuance.

TOKEN NODES

There will be special nodes for aggregating the OP_CODE data from the block chain. These nodes act similar to how a blockchain explorer operates, parsing each block and searching for relevant information to add to the “Token Chain”. The token nodes store the relevant data in a manner which is easier to query. The token nodes also have an API interface in which to query for information about the token chain.

We will run a minimum of 3 token nodes in a quorum configuration, which means that if any node gets out of sync, it is taken offline until it can resync it’s data. You may also run a token node for personal use if you do not want to use the ones provided by Pyrk.

TOKEN NODE API

Token nodes will provide API service for Token information. The API requests for the wallet UI and CLI will be built into the client, so it is not necessary for you to learn how to use the API. It is available for other developer usage.

WALLET UI & CLI

Additional commands will be added to the wallets to provide an easy to use interface for creating and managing your tokens. Additionally, the wallet will set a default address for you to use when creating, sending, and receiving tokens. It is best to use only one address for this purpose, as you will not be able to create a send transaction from multiple addresses, only one address at a time.

```
{ “tokens", “token_getbalance", &token_getbalance, false,  {"tokenid"} },
{ “tokens", “token_getbalances", &token_getbalances, false,  {} },
{ “tokens", “token_send", &token_send, false,  {"tokenid","address","amount"} },
{ “tokens", “token_listtransactions", &token_listtransactions, false, {"tokenid"} },
{ “tokens", “token_gettransaction", &token_gettransaction, false, {"transactionid"} },
{ “tokens", “token_addmeta", &token_addmeta, false,{"tokenid","metacode","metadata"} },
{ “tokens", “token_burn", &token_burn, false, {"tokenid","amount"} },
{ “tokens", “token_pause", &token_pause, false, {"tokenid"} },
{ “tokens", “token_resume", &token_resume, false, {"tokenid"} },
{ “tokens", “token_getinfo", &token_getinfo, false, {"tokenid"} },
{ “tokens", “token_getmeta", &token_getmeta, false, {"tokenid","metacode"} },
{ “tokens", “token_newowner", &token_newowner, false, {"tokenid","address"} },
{ “tokens", “token_authmeta", &token_authmeta, false, {"tokenid","address"} },
{ “tokens", “token_revokemeta", &token_revokemeta, false, {"tokenid","address"} },
{ “tokens", “token_create", &token_create, false, {"ticker","name", "genesisamount",
"documenturi", "logouri"} },
```