**Explanation of the Wormhole Agreement**

The WormHole protocol uses the bitcoin cash transaction as a carrier; the special opcode OP\_RETURN in the bitcoin cash script is used to append the WormHole protocol to the opcode. In general: the WormHole transaction is a special kind of bitcoin cash transaction that uses the same security and validation model as the bitcoin cash transaction.

The WormHole full-node client is a superset of the bitcoin-abc core client and has two functions: bitcoin cash protocol + WormHole protocol.

The processing flow of the WormHole transaction in the core client: After receiving the block---》bitcoin cash module confirms the legality of the transaction---》WormHole module to confirm the legality of the WormHole transaction.

The WormHole protocol uses an account model. Each bitcoin cash address is an account, and each account address can contain multiple tokens.

**Protocol Field Definition**

**Field: Transaction version**

Description: The version of each transaction type is monotonically increasing and independent of each other

Byte: unit16\_t; 2 bytes

**Field: Transaction type**

Description: The current function offered by Wormhole

Byte: unit16\_t; 2 bytes

Valid Values:

* 0: token transfer
* 1: participate in crowdfunding
* 3: token airdrop
* 4: transfer all available token in the account
* 50: create a fixed amount of token
* 51: Create a crowdfunding token
* 53: Forced closure of crowdfunding
* 54: Create a manageable token
* 55: Additional issuance token
* 56: Destroy the token
* 68: Obtain the base currency of the system (WHC)
* 70: Modify the issuer of the token

**Field: Ecosystem**

Description: The ecosystem in which the token is located;

Byte: uint8\_t; 1 byte

Valid value: The current valid value is 1;

**Field: Number of coins**

Description: amount of tokens stored in the protocol. Examples are as follows:

* For a Token with a precision after the decimal point, the value of the field is divided by the Precision (Transaction type) to identify the number of Tokens; (for a Token with a precision of 1, the value stored in the field is 10, and the amount of the Token is 1)
* For an indivisible Token (the precision Transaction type is 0), the value of this field indicates the amount of the Token; (if the value stored in the field is 100, the amount of the Token is 100)

Byte: int64\_t; 8 bytes Valid values: 1-9, 223, 372, 036, 854, 775, 807

* for a Token with a precision of 8, amount range: 0.00000001 - 92,233,720,368.54775807
* for a Token with a precision of 7, amount range: 0.0000001 - 922,337,203,685.4775807 ...
* for the indivisible Token (precision is 0), amount range: 1 - 9,223,372,036,854,775,807

**Field: Currency identifier**

Description: tokenID in the system

Byte: uint32\_t; 4 bytes

Valid values: 1, 3-2, 147, 483, 647;

0: BCH; 1:WHC; [3, 2,147,483,647] The TokenID that is subsequently created, monotonically increasing.

**Field: Property Precision**

Description: precision identification of assets in the Wormhole protocol

Byte: uint16\_t; 2 bytes

Valid values: range [0, 8]; respectively indicate decimal point precision

* 0: means no decimal point
* 1: indicates a precision of 1 after the decimal point
* .....
* 8: indicates a precision of 8 after the decimal point

**Field: Integer-one byte**

Description: used as a multiplier or as a parameter for other calculations

Byte: uint8\_t; 1 byte

Valid values: 0-255

**Field: UTC Datetime**

Description: Unix timestamp

Byte: uint64\_t; 8 bytes

**The specification for Wormhole protocol**

WormHole protocol requirements for transaction input and output order

In the WormHole system, the requirements for the order are mainly divided into two categories:

|  |
| --- |

|  |  |  |
| --- | --- | --- |
| Category | txin | txout |
| WHC's issuance transaction (68) | The first transaction input (index 0) is the sender | The first transaction output (index 0) is the combustion output, and the second output (index 1) is the WormHole protocol. |
| Other types of transaction | The first transaction input (index 0) is the sender | The last transaction output (the largest index) is the receiver |

**Issuance of WHC (68)**

The WHC is obtained by burning the BCH, and the exchange ratio of the BCH to the WHC is 1:100; the WHC can be obtained by transmitting the BCH to the specified combustion address. Burning address:

mainnet: bitcoincash:qqqqqqqqqqqqqqqqqqqqqqqqqqqqqu08dsyxz98whc

test net: bchtest:qqqqqqqqqqqqqqqqqqqqqqqqqqqqqdmwgvnjkt8whc

In the main net, after the 1000 blocks are confirmed, the WHC will arrive; in the test net, after the three blocks are confirmed, the WHC will arrive.

Regression test network: bchreg:pqqqqqqqqqqqqqqqqqqqqqqqqqqqqp0kvc457r8whc

In the main network, after the 1000 blocks are confirmed, the WHC will arrive; in the test network, after the three blocks are confirmed, the WHC will arrive.

Regression test network: bchreg:pqqqqqqqqqqqqqqqqqqqqqqqqqqqqp0kvc457r8whc

Examples are as follow：

Assume that when the main chain height is h1 = 55, the 1BCH is burned from the address1 address to obtain the WHC; at the main chain height of 1054, the address1 will receive 100WHC; after the block height of 1054, it can take the address1 to burn the received 100WHC. .

1BCH = 100WHC; 1WHC = 10000,0000C

The trading specification for obtaining WHC by burning BCH: Provisions for transaction input: The WormHole engine defaults to the first transaction input (ie, the index is 0) as the account address for obtaining the WHC.

The rules for trading output: WormHole engine default: the first transaction output must be the combustion output (ie index 0); the second output is the data of the WormHole protocol; the third output is the redundant BCH redemption address output (this The output can be ignored); a correct combustion transaction with a minimum of two outputs.

Provision for the amount of combustion: a minimum of 1 BCH.

The transaction specification mainly involves: the order of transaction input and output, the amount of combustion, and the WormHole load data.

WormHole This type of transaction protocol field is as follows:

|  |  |  |
| --- | --- | --- |
| Field | Type | Example |
| Transaction version | Transaction version | 0 |
| Transaction type | Transaction type | 68 |

In addition to the field restrictions of the WormHole protocol, WormHole transactions that are also invalid in the following cases

* The output of the burning BCH is not at the position where the index is 0.
* The index 1 position is not WormHole's combustion protocol data.
* The amount of BCH burned is less than 1BCH

WormHole payload data: 6a080877686300000044 explained as follows:

6a: OP\_RETURN; 08: The length of the payload data of the WormHole protocol;

08776863: The magic of the WormHole protocol

0000: version

0044: Transaction type

Example transaction (test chain): 6b8b98f7248f96b168b8127e3eb31cabb00fd6a5eed5df6d012816d7b4335974

**Token Transfer (0)**

Transfer a certain amount of Token from the specified account address to another account address

Note: The last output must be the recipient of the Token and the first input must be the sender of the Token

For the WHC transferring: Number of Coins, The unit identified by this field is WHC (eg: transfer 1 WHC, the value of this field is 1)

Specification of the transfer transaction:

* Transaction input: \*\*Wormhole\*\* engine defaults the first transaction input (index 0) is the account address (ie sender) that sends the token;
* Transaction output: \*\*Wormhole\*\* （search）engine defaults the last transaction output (the largest index) is the account address (ie the recipient) that receives the token;
* Transfer Amount: The amount of Token transferred by the sender to the recipient. The sender account must have sufficient assets.
* TokenID transferred: The Token must exist in the system.
* The transaction specification mainly involves: \*\*Transaction I/O order, transfer amount, TokenID\*\*

**Wormhole** protocol fields for this type of transaction are as follows:

Filed | Type | Example|

----| ---| ---|

Version (Transaction version) | Transaction version | 0

Transaction type | Transaction type |68

TokenID(Property ID) | Currency identifier| 1(WHC)

Amount to transfer|Number of Coins|1(identifies transferring 1WHC)

In addition to the field restrictions of the Wormhole protocol, Wormhole transactions are also invalid in the following cases:

* The amount of the transfer exceeds the amount of Token than the sender can use
* Transfer Token does not exist
* TokenID transferred is 0 (BCH); this transaction type does not support BCH transfers.

**Wormhole** payload data: 6a140877686300000000000000010000000005f5e100

The note is as follows:

6a: OP\_RETURN; 14: Data length of Wormhole protocol; 08776863: Magic 0000 of Wormhole protocol: Version 0000: Transaction type 00000001: TokenID 0000000005f5e100: Transfer amount (1WHC)

Example transaction:

d9dd78f140691dd946175696bd312dfcfac4c5cc5c4f5eb1143dfde0db83c25e

**Participate in crowdfunding (1)**

Participating in a specified crowdfunding transactions;

Note: **The last output must be the address of the crowdfunding issuer. The first input must be the WHC sender**

Participation in the crowdfunding transaction:

* Transaction input: \*\*Wormhole\*\* (search) engine defaults the first transaction input (index is 0) is the account address (ie sender) that sends the token;
* Transaction output: \*\*Wormhole\*\* engine defaults the last transaction output (the largest index) is the account address (ie the recipient) that receives the token;
* Transfer Amount: The amount of Token transferred by the sender to the recipient. The sender account must have sufficient assets.
* TokenID transferred: The Token must exist in the system.
* The transaction specification mainly involves: \*\*Transaction I/O order, transfer amount, TokenID\*\*

**Wormhole** protocol fields for this type of transaction are as follows:

Filed | Type | Example|

----| ---| ---|

Transaction version |Transaction version | 0

Transaction type |Transaction type | 1

TokenID(Property ID) | Currency identifier| 1(WHC)

Amount of participation in crowdfunding (Amount to transfer)|Amount of Coins|1 (token used to purchase 1WHC)

In addition to the field restrictions of the Wormhole protocol, Wormhole transactions are also invalid in the following cases.

* The amount of participation in crowdfunding exceeds the amount of Token that the sender can use.
* There is no active crowdfunding at the crowdfunding address
* The amount of participation in crowdfunding is not enough to buy the minimum precision of crowdfunding tokens.

**Wormhole** payload data: 6a140877686300000001000000010000000005f5e100

**Token Airdrop (3)**

The specified amount of token (ID1) is dropped to the holder of a certain token (ID2) (except the sender), and the amount (ID1) obtained by all the addresses accepted by the airdrop is allocated according to the token (ID2) ratio. . Note: The first input must be the sender of the airdrop

Airdrops will charge a certain fuel fee, which is denominated in WHC. The handling fee for each airdrop is: the number of people receiving airdrops\* system unit price (1C);

This type of transaction protocol field is as follows:

Filed | Type | Example|

----| ---| ---|

Transaction version |Transaction version | 0

Transaction type |Transaction type | 13

TokenID(Property ID) | Currency identifier| 1(WHC)

Amount of token airdrop (Amount to transfer)| 15000000000(150WHC)

TokenID(Property ID) | Currency identifier| 1(WHC)

In addition to the field restrictions of the WormHole protocol, WormHole transactions that are also invalid in the following cases

* The amount available on the sender's address is less than the amount of the airdrop
* The specified token does not exist, or the tokenID is 0.
* There is not enough WHC to pay for fuel on the sender's address.
* The sender address contains all destination tokens

WormHole payload data: 6a18087768630000000300000001000000037e11d600 is explained as follows:

6a: OP\_RETURN; 18: length of the payload data of the WormHole protocol;

08776863: The magic of the WormHole protocol

0000: version

0003: Transaction type

00000001: Airdrop tokenID

000000037e11d600: Airdrop amount (1.5WHC)

00000001: destination tokenID

Example transaction: 1db5d3b5c094e0d839465a1623d1b29baccb03e1bc2d817c6ee87d8b7ca4ebc4

**Transfer all available tokens for the account (4)**

Transfer all tokens of one account address (A) to another account address (B), where the asset does not contain BCH. Note: The last output must be the recipient of the token, the first input must be the sender of the token

WormHole This type of transaction protocol field is as follows:

Filed | Type | Example|

----| ---| ---|

Transaction version |Transaction version | 0

Transaction type |Transaction type | 4

Ecosystem | Ecosystem | 1(WHC)

WormHole payload data: 6a09087768630000000401 is explained as follows:

6a: OP\_RETURN; 09: \*\*WormHole\*\* protocol data length;

08776863: The magic of the WormHole protocol

0000: version

0004: Transaction type

01: Ecosystem

Example transaction: 6b6b04aed7ed091a9b1f89b9517064cb16cd5032f1b0994d93b6db2a5d20e871

**Create a fixed number of tokens (50)**

Once the transaction is confirmed valid by WormHole, the creator's address will get contain these tokens.

Except for the field restrictions of the WormHole protocol, the transaction is considered to be the correct WormHole transaction if the following conditions are met.

* Property Name field cannot be NULL
* Ecosystem field must be 1
* The total token amount created by Number of coins is within the range
* Property precision asset precision is within range
* WormHole This type of transaction protocol field is as follows

WormHole This type of transaction protocol field is as follows:

Filed | Type | Example|

----| ---| ---|

Transaction version |Transaction version|0

Transaction type |Transaction type|50 Ecosystem|Ecosystem|1

Property Precision| Property Precision |1

Previous Property ID|Currency identifier|0

Property Category|String null-terminated|""

Property Subcategory|String null-terminated|""

Token Name(Property Name)|String null-terminated|"ludete"

Token URL(Property URL)|String null-terminated|""

Token Description Data(Property Data)|String null-terminated|""

Total amount of generated (totaltokens) | Number of coins| 10000

WormHole payload data: 6a41087768630000003201000100000000746573740074657374320066697865642d746f6b656e0066697865642d746f6b656e006d7964617461000000000005f5e100

Explanations are as follow:

6a: OP\_RETURN; 41: length of the payload data of the WormHole protocol;

08776863: The magic of the WormHole protocol

0000: version

0032: Transaction type

01: Ecosystem

0001: token precision

00000000: Front TokenID

.... : The middle is some custom data

0000000005f5e100: The amount of token issued

Example transaction: acec40a5ca1efed57a311b7dbb9324e753e7f5dd2395b3825163672b5edfc809

**Create Token can be crowdfunded (51)**

Create a token can be crowdfunded, and once the transaction is validated by the Wormhole node, the sender's address will have an active crowdfunding.

Note: Current crowdfunding is only allowed to use WHC to participate

When the following happens, crowdfunding will be permanently closed:

* When the blockchain's Tip block timestamp is greater than or equal to the crowdfunding Deadline, the crowdfunding will be permanently closed.
* Crowdfunding is manually forced to close
* The crowdfunding amount is sold out.

When crowdfunding is closed, unsold crowdfunding tokens are counted in the issuer's account address.

Any account address is allowed to have only one active crowdfunding token at a given time, avoiding crowdfunding participants who need to specify which crowdfunding to participate in the account.

In addition to the field restrictions of the Wormhole protocol, the transaction will only be considered to be correct Wormhole transaction if the following conditions are met.

* Property Name field cannot be NULL
* Ecosystem field must be 1
* Property Type field must be 1
* Currency Identifier Desired: Token being crowdfunded is only allowed to be WHC(1)
* Deadline: deadline of crowdfunding must be later than the start time
* Crowdfunding buyer must not be the crowdfunding participant

When other accounts initiate transactions to participate in the crowdfunding, once the transactions participating in the crowdfunding are confirmed by the Wormhole protocol, the purchased crowdfunding will be immediately counted into in the available funds of the participant's account, and the participants can transfer the purchased crowdfunding, sale, or other purposes; at the same time, the funds for participation in crowdfunding (WHC) are immediately counted into the funds available for the crowdfunding issuer's account, and the crowdfunding issuer can use the funds.

In order to encourage users to participate in crowdfunding as early as possible, early participants can be rewarded by setting up an early bird reward mechanism. This early bird reward will decrease linearly with the time when the crowdfunding starts, until the crowdfunding deadline, the early bird incentive will be 0.

In order to encourage crowdfunding participants to participate as early as possible, early participants can be rewarded by setting up an early bird reward mechanism. This early bird reward linearly decreases with the start of crowdfunding and the early bird rewards will be 0 until the deadline.

The percentage of early bird reward is calculated as follows:**((Crowdfunding deadline - the time of block the transaction located) / number of seconds per week)) \* Early bird reward**

Examples are as follows:

Two Token exchange ratios: 1WHC=100HLT; Early bird reward mechanism is 10%, such as purchasing 1WHC: The crowdfunding participant's transaction is two weeks before its deadline, buyers will get 20% extra token (120HLT). If it is 1 week and a half before the deadline, buyers will get 15% extra Token (115HLT).

The Wormhole client guarantees that the number of tokens credited to the crowdfunding participant plus the number of tokens in the issuer's account does not exceed the total circulation. When a transaction involving crowdfunding is insufficient to purchase the minimum precision of the Token be crowdfunded, the crowdfunding transaction fails and the funds are returned to the participants.

**Wormhole** protocol fields for this type of transaction are as follows:

Filed | Type | Example|

----| ---| ---|

Transaction version |Transaction version|0

Transaction type |Transaction type|50

Ecosystem|Ecosystem|1

Token Property Pricision|Pricision|1

Previous Property ID|Currency identifier|0

Token Category(Property Category)|String null-terminated|""

TokenSubcategory|String null-terminated|""

Token Name(Property Name)|String null-terminated|"ludete"

Token URL(Property URL)|String null-terminated|""

Token Description Data(Property Data)|String null-terminated|""

TokenID for crowdfunding ( Currency Identifier Desired) |Currency identifier|1

Number Properties per Unit Invested|Number of Coins|100 (Token: Token be Crowdfunded)

Crowdfunding Deadline|Deadline|Unix Timestamp

Percentage of Early Bird Incentives (Early Bird Bonus %/Week)|Integer one-byte | 10

Undefined |Integer one-byte | must be 0

Total amount of crowdfunding (totaltokens) | Number of coins| 10000.8

**Wormhole payload data**: 6a4c6c08776863000000330100026974636f696e5f3000687474703a2f2f7777772e62616964752e636f6d0000 00000001 00000002540be400 000000005b881480 0a 00 000000000bebc200

The note is as follows:

6a: OP\_RETURN; 4c: OP\_PUSHDATA1; 6c: Data length of Wormhole protocol; 08776863: Magic 0000 of Wormhole protocol: Version 0033: Transaction type 01: Ecosystem 0002: Token category 000000000: Pre-TokenID .... : *The middle is some Custom Data* 00000001: Crowdfunding TokenID 00000002540be400: Conversion ratio 000000005b881480: Crowdfunding Deadline 0a: Percentage of Early Bird Incentives 00: Undefined filed 000000000bebc200 : Total amount of crowdfunding

example transaction: e00b24013f4ef929dd554c3233c5f7df60e6249eab395b8c7df60e6249eab395b8c2540f01486470cb9

**Stop crowdfunding (53)**

When the crowdfunding is active (i.e. raising funds), and the project team needs to stop the crowdfunding in order to keep the value of the token (or have other reasons to do so); it can be done by creating such a transaction and broadcast it to the Blockchain. The shutdown of crowdfunding will not affect the early bird bonus who already purchased the token.

Except for the field restrictions of the WormHole protocol, WormHole transactions are also invalid in the following cases:

• Shutting down a crowdfunding that has been closed

• The sender of this transaction is not the issuer of the crowdfunding token

WormHole’s stop crowdfunding transaction protocol field:

Filed | Type | Example|

Transaction version |Transaction version | 0

Transaction type|Transaction type| 53

TokenID (Property ID) |Currency identifier |9

Effective Wormhole payload data: 6a0c087768630000003500000005

Explanation are as follows:

6a: OP\_RETURN; 0c: length of the payload data of the WormHole protocol;

08776863: The magic of the WormHole protocol

0000: version

0035: Transaction type

00000005: tokenID

Example trading: f40861a873a6fafadc9bc252b433268e780f232a53c4ed324ab5919d1cd23c01

**Create manageable token (54)**

Creating a token that can be managed by the issuer. The address of the first transaction input (index 0) is the issuer of the token. At this time, the issuer's address has 0 available balance of the token, and two additional transaction types are used to issue a specific amount of the token (issue additional token), or destroy a specific amount of the token (destroy token).

Except for the field restrictions of the WormHole protocol, the WormHole transaction is considered valid only if the following conditions are met:

• Property Name field must not be NULL

• Ecosystem field must be 1

WormHole’s create manageable token transaction protocol field:

Filed|Type|Example

Transaction version| Transaction version| 0

Transaction type| Transaction type|50

Ecosystem|Ecosystem|1

Token type (Property Type)|Property Type|1

Previous tokenID (Previous Property ID)|Currency identifier|0

Toke category (Property Category)|String null-terminated|""

Token subcategory (Property Subcategory)|String null-terminated|""

Token name (Property Name)|String null-terminated|"ludete"

Token URL (Property URL)|String null-terminated|""

Token description data (Property Data)|String null-terminated |""

WormHole’s valid payload data: 6a3608776863000000360100010000000048656c6c6f00576f726c64006c7564657465007777772e6c75646574652e636f6d00796f6e6700

Explanation are as follows:

6a: OP\_RETURN; 36: The data length of WormHole protocol

08776863: Magic of WormHole protocol

0000: Version

0036: Transaction type

01: Ecosystem

0001: Token type

00000000: Previous token ID

.... : Customized data

Example of transaction：b85ffb2d38339b4432cd8dfe50861e05376665f1ffe08c4824dea934aadb20e1

**Issue additional token (55)**

This transaction type is used to service the manageable token by issuing additional token amounts. After the issuance transaction is confirmed, the additional token amount will be added to the recipient's available balance. Note: After the token is created, the initial amount is 0; the maximum legal amount is: (1 << 63) -1

Except for the field restrictions of the WormHole protocol, WormHole transactions are also invalid in the following cases:

• The additional token category is not a manageable type of token

• The sender of the transaction is not the token issuer

• The sum of the cumulative tokens exceeds the maximum legal amount.

WormHole’s issue additional token transaction protocol field:

Filed|Type|Example

Transaction type|transaction type|55

tokenID (Property ID)|Currency identifier |5

Additional issued amount (Number Properties)|Number of coins|1000

Additional issuance message (Memo)|String null-terminated|"reward"

WormHole’s valid payload data：6a1d08776863000000370000000c00000000000027107061792062696c6c00

Explanation are as follows：

6a: OP\_RETURN; 1d: The data length of WormHole protocol;

08776863: Magic of WormHole protocol

0000: Transaction version

0037: Transaction type

0000000c: tokenID

0000000000002710: Amount of additional issued token

.... : Customized data

Example of transaction：6a1d08776863000000370000000c00000000000027107061792062696c6c00

**Revoke token (56)**

This transaction type is used to service the manageable token by destroying token amounts. After the destroying transaction is confirmed, and the total token amount will also be deducted from the system.

Except for the field restrictions of the WormHole protocol, WormHole transactions are also invalid in the following cases:

• The destroyed token is not a manageable type of token

• The sender of the transaction is not the token issuer

• The number of destroyed tokens exceeds the amount of tokens available for the current account

WormHole’s destroy token transaction protocol field:

Filed|Type|Example

Transaction version|Transaction version|0

tokenID (Property ID)|Currency identifier |5

Destroyed amount of tokens| (Number Properties) Number of coins|1000

Destroy message (Memo)|String null-terminated|"spend"

WormHole’s valid payload data：6a1508776863000000380000000c0000000000000bb800

Explanation are as follows：

6a: OP\_RETURN; 15: The data length of WormHole protocol；

08776863: Magic of WormHole protocol

0000: Transaction version

0038: Transaction type

0000000c: tokenID

0000000000000bb8: Destroyed amount

.... : Customized data

Example of transaction：d5966fd7568b29b19ba5400d94e4fa8adbc91040fd0261f1209db163518a0327

**Change the issuer of the token (70)**

This transaction type is used to change the issuer of a token which is already issued in the system.

Conditions for transaction input: The first transaction input is the original issuer of the token

Conditions for transaction output: The last transaction output is the new token issuer

Except for the field restrictions of the WormHole protocol, WormHole transactions are also invalid in the following cases:

* There is no such asset
* Any account of rhe sender or receiver is active in crowdfunding will fail to transfer the asset.
* The sender is not the original issuer
* The address of sender and recipient address are the same;
* The recipient is empty;

WormHole’s change issuer of token transaction protocol field:

Filed|Type|Example

Transaction version|Transaction version|0

Transaction type|Transaction type|55

tokenID (Property ID)|Currency identifier|5

WormHole’s valid payload data:：6a0c08776863000000460000000c

Explanation are as follows：

6a: OP\_RETURN; 15: The data length of WormHole protocol；

08776863: Magic of WormHole protocol

0000: Transaction version

0038: Transaction type

0000000c: tokenID

Example of transaction：ce6877f4da9e627741cf087aa777917f5ecf5fb8ff47037b211140b68155d9eb