1.0.9 Release Notes

Bitcoin SV Node software – Upgrade to v1.0.9 Release

Version 1.0.9 release is a recommended upgrade from version 1.0.8; This new version brings improved safe mode processing, webhook notifications, block double spend P2P notification messages and the possibility to freeze transaction outputs that are the target of double spends.

Content details listed below:

1. **New block double spend P2P notification message.**
2. **Improved safe mode processing including new RPCs to manually control safe mode.**
3. **Webhook notification of competing chains and double spends.**
4. **Possibility to freeze transaction outputs.**
5. **Performance improvements to transaction chain processing.**
6. **STN Reset**

Response to Block Withholding Attacks

Background

In the past few months, there have been several attempted attacks on the BSV network. The attacks involve mining a hidden alternate chain (in at least one case > 80 blocks in length) and then posting it to the blockchain at once so it became the new main chain. The new chain replaces some of the transactions in the original chain with double spends. Exchanges typically require a certain number of block confirmations (say 50 blocks have been built on the attempted transactions) before coin deposits or withdrawals into their exchange are valid, *for example*, to prevent an attacker from depositing double-spent BSV coins and immediately trade or cash out his position in the selfish chain after previously doing the same thing on the main chain.

The BSV blockchain and other PoW blockchains are vulnerable to these types of attacks because of the large amounts of roaming hash-power available that can be used on competing networks.

Our response to these attacks has included steps to:

1. Put global 24 hour/7 day a week monitoring in position and:
    - notify miners of long, competing chains when they appear.
    - notify exchanges and application providers of double spends in the new chain.
2. Enable honest miners to invalidate the block at the base of the selfish chain using ***invalidateblock***, and then mine on the original honest chain until it is restored as the main chain.

Notification of change of safe mode (competing chains)

Exchanges and miners will be notified if a suspicious event (most likely an attack) occurs. The trigger is the existence of recent, long, competing chains.

Notification about a block reorganisation is part of the safe mode notification.

The Bitcoin SV Node software currently contains “safe-mode” logic to detect competing chains and de-activate wallet RPC calls (put into safe-mode). The safe-mode logic has been updated so that chain detection is configurable (allowing for users to customise their sensitivity to suspicious events) and triggers notifications via webhooks to exchanges/miners/application providers.

Notification of double spends

Exchanges and miners will be notified of recent double spends in competing chains of the BSV network.

Blockchain monitoring applications may send a DSD P2P notifications message to a BSV node (meaning a mining node), which will then relay the message to other nodes. The message contains a double spend proof (Merkle proof of the inclusion of transactions in a block). Notifications are via webhooks.

Freezing of double spends

A BSV node (meaning a mining node) can now "freeze" specific TXOs (e.g. used by double spends).

If a policy freeze is applied to the TXO, the node will not accept transactions that spend the TXO, but it will accept external blocks containing transactions that spend the TXO.

Specific Changes

Interface with DS Detector

The node will receive DSD P2P notification message that a double spend has been detected. On receipt of a DSD message, (a) the DSD message is verified/validated (to ensure that is not a malicious fake), (b) the DSD message is relayed to other nodes, and (c) webhooks are used to notify users.

Note that there is no need to inform peers about every double-spend contained in every block, it is sufficient for a notification to just contain the details of a single transaction from each block containing a conflict.

DSD P2P message

The format of the new DSD P2P message is

Field size	Description	Data Type	Comments
2	version	uint16_t	Versioning information for this message. Currently can only contain the value 0x0001.
1+	block count	varint	The number of blocks containing a double spend transaction this message is reporting. Must be >= 2.
variable	block list	block_details[]	An array of details for blocks containing double spend transactions

block_details:

size	Description	Data Type	Comments
1+	header count	varint	The number of following block headers (may not be 0).
variable	header list	block_header	An array of unique block headers containing details for all the blocks from the one containing the conflicting transaction back to the last common ancestor of all blocks reported in this message. Note that we don't actually need the last common ancestor in this list, it is sufficient for the last header in this list to be for the block where this fork begins, i.e.; the **hashPrevBlock** field from the last block header will be the same for all the blocks reported in this message.
variable	merkle proof	merkle_proof	The transaction contents and a proof that the transaction exists in this block. Follows TSC standard.

block_header:

Field size	Description	Data Type	Comments
4	version	int32_t	Block version information.
32	hash previous block	char[32]	Hash of the previous block in the fork.
32	hash merkle root	char[32]	The merkle root for this block.
4	time	uint32_t	Timestamp for when this block was created.
4	bits	uint32_t	Difficulty target for this block.
4	nonce	uint32_t	Nonce used when hashing this block.

The DSD P2P message specification can be found at https://github.com/bitcoin-sv-specs/protocol/blob/master/p2p/DSD_P2P.md

Webhook DSD notification message

The node will use webhooks to notify listeners that new double spends has been detected on alternate chains on the blockchain. The recipient of the webhook message can be configured using the new command line parameter, ***dsdetectedwebhookurl***.

The DSD notification message template is shown below

```
{
   "version" : number,
   "blocks : [
      {

         "divergentBlockHash" : string,
         "headers" : [
            {
               "version" : number,
               "hashPrevBlock" : string,
               "hashMerkleRoot" : string,
               "time" : number,
               "bits" : number,
               "nonce" : number
            }
         ],
         "merkleProof" : {
            "index" : number,
            "txOrId" : string,    // Full transaction, serialised, hex-encoded
            "targetType": "merkleRoot",
            "target" : string,    // Merkle-root
            "nodes" : [ "hash", ... ]
         }
      },
      ...
   ]
}
```

This notification message is documented in docs/web_hooks.md in the distribution.

The **getblockheader** RPC has been updated to report on the valid/invalidated status of the block.

Update of Safe-Mode Trigger Methodology

Safe-mode disables node wallet functionality so that users cannot spend coins. Safe-mode is triggered when a long competing (parallel) block chain is detected. *i.e.* an attack chain is found. Webhooks are used to notify users when safe-mode status has been changed.
This update also modifies the logic that triggers safe-mode.

This release introduces 3 new command line parameters

1. **safemodemaxforkdistance**, Default = 1000 (7 days of mining)
2. **safemodeminforklength**, Default = 3
3. **safemodeminblockdifference**, Default = -72 (12 hours of mining below current tip)

Safe mode is automatically triggered if all of these criteria are satisfied:

- The distance between the current tip and the last common block header of the fork is smaller than the **safemodemaxforkdistance**.
- The length of the fork is greater than **safemodeminforklength**.
- The total proof of work of the fork tip is greater than the minimum fork proof of work (POW). The minimum fork POW is calculated relative to the active chain tip using this formula: *<total-proof-of-work-active-chain>* + **safemodeminblockdifference** * *<proof-of-work-of-the-active-tip>.* Safe mode is activated if the fork height is bigger than <height-active-chain-tip> + **safemodeminblockdifference.** Note that the negative value of **safe modeminblockdifference** means that we will activate the safe mode for forks with tips below active chain tip.

the first condition from the current implementation is retained, but triggering occurs before the competing fork surpasses the active chain. Safe mode is activated whenever the competing chain tip approaches the main chain tip closer than **safemodeminforklength** or it is ahead of the main chain tip.

Webhook Safe Mode notification message

The node will use webhooks to notify listeners if the node enters or leaves safe-mode or one of the following events occur while in safe mode.

- the non-main chain is extended
- a block is fetched for the tip of non-main chain.
- a change in the validity of a chain (e.g. triggered by calls to **invalidateblock**)
- a blockchain reorganisation

Notifications will not occur during initial block download or reindex even if safe mode conditions exist.

A users who is only interested in the current safe mode state can check the **safemodeenabled** field in the notification message (see below).

The recipient of the webhook message can be configured using the new command line parameter ***safemodewebhookurl***.

The safe mode notification message template is shown below.

```
{
   "safemodeenabled": boolean,          // Is the node in the safe mode.
   "activetip": {                       // Tip of the main chain.
      "hash": string,                   // Block hash.
      "height": number,                 // Block height.
      "blocktime": string,              // Human readable, UTC, block time.
      "firstseentime": string,          // Human readable, UTC, time when the node first received header.
      "status": "active"                // Status of the block. Possible values: active, invalid
                                        //     headers-only, valid-fork, and valid-headers.
   },
   "timeutc": string,                   // Human readable, UTC, time of creation of this message.
   "reorg": {                           // Information about possible reorg.
      "happened": bool,                 // Indicates if an reorg happened.
      "numberofdisconnectedblocks": number, // Number of blocks disconnected in reorg.
      "oldtip": {                       // Information about old active tip, "null" if an reorg did not happened.
         "hash": string,
         "height": number,
         "blocktime": string,
         "firstseentime": string,
         "status": string
      }
   },
   "forks": [                           // List of forks that are triggering the safe mode.
      {
         "forkfirstblock": {            // Root of the fork, this block's parent is on the main chain.
            "hash": string,
            "height": number,
            "blocktime": string,
            "firstseentime": string,
            "status": string
         },
         "tips": [                      // List of tips of this fork.
            {
               "hash": string,
               "height": number,
               "blocktime": string,
               "firstseentime": string,
               "status": string
            },
            ...
         ],
         "lastcommonblock": {           // Block on the main chain which is parent of the "forkfirstblock"
            "hash": string,
            "height": number,
            "blocktime": string,
            "firstseentime": string,
            "status": string
         },
         "activechainfirstblock": {     // Block on the main chain which is child of the "lastcommonblock"
            "hash": string,
            "height": number,
            "blocktime": string,
            "firstseentime": string,
            "status": string
         },
      },
   ...
   ]
}
```

This notification message is also documented in docs/web_hooks.md in the distribution.

New RPC functions to enable/disable Safe-Mode

The RPC function allows a user to manually override safe-mode without restarting the node.

This release introduces 3 new RPCs.

1. **ignoresafemodeforblock <block_hash>** - The specified block and all its descendants will be ignored when calculating criteria for entering the safe mode.
2. **reconsidersafemodeforblock <block_hash>** - The specified block and all its descendants will be considered when calculating criteria for entering the safe mode.
3. **getsafemodeinfo** - Detailed information on the safe mode status.

The effect of the RPCs is not preserved across node restarts.

Support for Transaction Output Freezes

A node can now "freeze" specific transaction outputs (TXOs). A freeze can be applied to both spent and unspent TXOs, and can be used to ensure that a node does not process a known double spend.

If a policy freeze is applied to the TXO, the node will not accept transactions that spend the TXO, but it will accept external blocks containing transactions that spend the TXO.

This release introduces new RPCs.

1. **addToPolicyBlacklist <funds : object>** - add TXO to policy blacklist (i.e. do not accept transactions that accesses this TXO, blocks containing transaction may be accepted if not in the consensus blacklist)
2. **removeFromPolicyBlacklist <funds : object>**- remove TXO from policy blacklist
3. **queryBlacklist** - Returns all frozen TXOs from the DB.
4. **clearBlacklists <removeAllEntries : boolean, default=true>** - Removes entries from the blacklist.

Performance Improvements

The release contains optimisation to the processing of long, complex chains of transactions under extreme loads.