


Block header

```
13  /** Nodes collect new transactions into a block, hash them into a hash tree,
14   * and scan through nonce values to make the block's hash satisfy proof-of-work
15   * requirements. When they solve the proof-of-work, they broadcast the block
16   * to everyone and the block is added to the block chain. The first transaction
17   * in the block is a special one that creates a new coin owned by the creator
18   * of the block.
19   */
20  class CBlockHeader
21  {
22  public:
23      // header
24      int32_t nVersion;
25      uint256 hashPrevBlock;
26      uint256 hashMerkleRoot;
27      uint32_t nTime;
28      uint32_t nBits;
29       uint32_t nNonce;  $2^{32}$ 个可能
```

source: bitcoin/src/primitives/block.h



Block headers are serialized in the 80-byte format described below and then hashed as part of Bitcoin's proof-of-work algorithm, making the serialized header format part of the consensus rules.

Bytes	Name	Data Type	Description
4	version	int32_t	The block version number indicates which set of block validation rules to follow. See the list of block versions below.
32	previous block header hash	char[32]	A SHA256(SHA256()) hash in internal byte order of the previous block's header. This ensures no previous block can be changed without also changing this block's header.
⇒ 32	merkle root hash	char[32]	A SHA256(SHA256()) hash in internal byte order . The merkle root is derived from the hashes of all transactions included in this block, ensuring that none of those transactions can be modified without modifying the header. See the merkle trees section below.
4	<u>time</u> 区块产生时间	uint32_t	The block time is a Unix epoch time when the miner started hashing the header (according to the miner). Must be strictly greater than the median time of the previous 11 blocks. Full nodes will not accept blocks with headers more than two hours in the future according to their clock.
4	<u>nBits</u> 目标值	uint32_t	An encoded version of the target threshold this block's header hash must be less than or equal to. See the nBits format described below.
⇒ 4	<u>nonce</u>	uint32_t	An arbitrary number miners change to modify the header hash in order to produce a hash less than or equal to the target threshold . If all 32-bit values are tested, the time can be updated or the coinbase transaction can be changed and the merkle root updated.

The hashes are in **internal byte order**; the other values are all in **little-endian order**.

source: bitcoin.org

