

Table of Contents

Model Detail	12
Class Model	12
<anonymous>	12
<anonymous>	13
<anonymous>	14
<anonymous>	14
<anonymous>	16
<anonymous>	18
<anonymous>	19
<anonymous>	20
AcceptedConnection	21
BanReason	22
BaseSignatureChecker	23
BaseSignatureCreator	24
BasicTestingSetup	26
BitcoinAddressCheckValidator	26
BitcoinAddressEntryValidator	27
BitcoinUnit	28
BitcoinUnits	28
RoleIndex	31
SeparatorStyle	32
Unit	33
BlockHasher	34
BlockMap	34
BlockStatus	34
CAccount	37
CAccountingEntry	38
CAddrDB	41
CAddrInfo	42
CAddrMan	46
CAddress	52
CAddressBookData	54
StringMap	55
CAffectedKeysVisitor	56
CAlert	57
CAmount	60
CAutoFile	62
CBanDB	65
CBanEntry	66
CBase58Data	68
vector_uchar	71
CBaseChainParams	72
Network	73

CBaseMainParams	74
CBaseRegTestParams	75
CBaseTestNetParams	75
CBaseUnitTestParams	76
CBasicKeyStore	77
CBigNum	80
CBitcoinAddress	81
CBitcoinAddressVisitor	83
CBitcoinExtKey	84
CBitcoinExtKeyBase	85
CBitcoinExtPubKey	86
CBitcoinSecret	86
CBlock	87
CBlockFileInfo	89
CBlockHeader	92
CBlockIndex..... <anonymous>	96 101
CBlockLocator.....	101
CBlockPolicyEstimator	103
TxStatsInfo	107
CBlockTemplate.....	108
CBlockTreeDB	109
CBlockUndo	112
CBloomFilter	113
CBufferedFile	117
CChain.....	121
CChainParams..... Base58Type	122 127
CCheckQueue	129
CCheckQueueControl	131
CCheckpointData	132
CCClientUIInterface..... MessageBoxFlags	134 135
CCoinControl	139
CCoins	141
CCoinsCacheEntry	145
Flags	146
CCoinsKeyHasher	146
CCoinsMap	147
CCoinsModifier	148
CCoinsStats	149
CCoinsView	152
CCoinsViewBacked	153
CCoinsViewCache	155
CCoinsViewDB	158
CCoinsViewMemPool	160

CCrypter	162
CCryptoKeyStore	164
CDB	167
CDBEnv	170
KeyValPair	173
VerifyResult	174
CDNSSeedData	175
CDataStream	176
allocator_type	183
const_iterator	184
const_reference	184
difference_type	184
iterator	184
reference	185
reverse_iterator	185
size_type	185
value_type	186
vector_type	186
CDiskBlockIndex	187
CDiskBlockPos	188
CDiskTxPos	191
CECKey	192
CExtKey	194
CExtPubKey	196
CFeeRate	198
CFlatData	201
CHMAC_SHA256	203
CHMAC_SHA512	205
CHash160	207
CHash256	208
CHashWriter	209
CIInPoint	211
CIInv	212
CKey	215
CKeyID	219
CKeyMetadata	219
CKeyPool	221
CKeyStore	223
CKeyingMaterial	225
CKeyingMaterial	226
CLevelDBBatch	226
CLevelDBWrapper	227
CMainParams	231
CMainSignals	232
CMasterKey	234
CMedianFilter	235

CMerkleBlock.....	237
CMerkleTx	239
CMessageHeader.....	242
<anonymous>	244
CMutableTransaction.....	245
CNetAddr	247
CNetCleanup	252
CNetMessage	253
CNoDestination.....	256
CNode.....	256
CNodeSignals.....	277
CNodeStateStats	278
CNodeStats	279
COrphan	283
COrphanTx	285
COutPoint	286
COOutput	288
COOutputEntry	290
CPartialMerkleTree	291
CPriKey	294
CPubKey.....	295
CRIPEMD160	299
CRPCCCommand	301
CRPCCConvertParam.....	302
CRPCCConvertTable	303
CRPCTable.....	304
CRecipient	305
CRegTestParams	306
CReserveKey	306
CReserveScript.....	308
CRollingBloomFilter	309
CSHA1	312
CSHA256.....	314
CSHA512.....	316
CScheduler	317
Function	320
CScript	320
CScriptCheck	325
CScriptCompressor.....	328
CScriptID	330
CScriptNum	331
CScriptVisitor	334
CSerActionSerialize	336
CSerActionUnserialize	336
CSerializeData.....	337
CService	337

CSignatureCache.....	341
sigdata_type.....	343
CSizeComputer.....	343
CSubNet	344
CTestNetParams	347
CTransaction	348
CTransactionSignatureSerializer	352
CTranslationInterface.....	356
CTxDestination	356
CTxIn	357
CTxInUndo.....	360
CTxMemPool	362
CTxMemPoolEntry.....	368
CTxOut	371
CTxOutCompressor.....	374
CTxUndo.....	376
CUnsignedAlert.....	377
CValidationInterface.....	380
CValidationState	382
mode_state	385
CVarInt.....	386
CVerifyDB	387
CWallet	388
MasterKeyMap	407
TxItems	407
TxPair.....	408
TxSpends	408
CWalletDB	408
CWalletKey	414
CWalletScanState.....	416
CWalletTx	418
CachingTransactionSignatureChecker	427
ChainCode.....	429
ChainCode.....	429
ChangeType	430
CoinControlTreeWidget	431
CombinerAll	431
result_type.....	432
CompareValueOnly.....	432
CryptedKeyMap	433
DBErrors	433
DummySignatureChecker	435
DummySignatureCreator	436
HTTPStatusCode.....	436
HelpMessageMode	438
KeyMap.....	439

LimitedString	439
ListenSocket	441
LocalServiceInfo	441
LockedPageManager	442
LockedPageManagerBase	444
Histogram	446
MacNotificationHandler	446
MapCheckpoints	447
MemoryPageLocker	447
MutableTransactionSignatureChecker	448
Network	449
NetworkStyle	450
Nodeld	452
PaymentRequestPlus	453
PlatformStyle	455
ProxyCredentials	457
RPCErrorCode	458
RecentRequestEntry	464
RecentRequestEntryLessThan	465
SSLIOStreamDevice	467
SSLVerifyError	468
ScriptError_t	469
ScriptMap	474
SecureString	475
SeedSpec6	475
TestChain100Setup	476
TestingSetup	477
TransactionDesc	479
TransactionFilterProxy	480
WatchOnlyFilter	483
TransactionRecord	484
Type	488
TransactionSignatureChecker	490
TransactionSignatureCreator	492
TransactionStatus	494
Status	496
TxConfirmStats	498
TxInputStream	502
TxPriority	504
TxPriorityCompare	504
UniValue	505
VType	513
WalletFeature	514
WalletModelTransaction	515
WatchKeyMap	518
WatchOnlySet	518

WinShutdownMonitor.....	519
arith_uint256	520
banmap_t.....	521
base_blob	522
<anonymous>	525
base_uint	525
<anonymous>	532
bignum_error	533
bloomflags	533
ecgroup_order.....	534
int16_t.....	535
int32_t.....	535
int64_t.....	536
int8_t.....	541
isminefilter.....	542
isminetype.....	542
jtokentype	543
leveldb_error	546
limitedmap	546
const_iterator	548
iterator	549
key_type	549
mapped_type	550
rmap_iterator	550
size_type	550
value_type	551
mapValue_t.....	551
mruset.....	552
const_iterator	555
iterator	555
key_type	555
size_type	556
value_type	557
opcodetype	557
proxyType	576
scriptnum_error.....	577
secp256k1_ecdsa_sig_t	577
secp256k1_ecmult_context_t	578
secp256k1_ecmult_gen_context_t.....	579
secp256k1_fe_storage_t.....	580
secp256k1_fe_storage_t.....	581
secp256k1_fe_t.....	582
secp256k1_fe_t.....	584
secp256k1_ge_storage_t.....	585
secp256k1_ge_t.....	586
secp256k1_gej_t.....	588

secp256k1_hmac_sha256_t.....	589
secp256k1_num_t.....	590
secp256k1_rfc6979_hmac_sha256_t.....	591
secp256k1_scalar_t.....	592
secp256k1_scalar_t.....	593
secp256k1_sha256_t.....	594
secure_allocator.....	596
base	597
const_pointer.....	597
const_reference	598
difference_type.....	598
pointer	599
rebind	599
other	599
reference	600
size_type	600
value_type.....	600
txnouttype	601
uint160	602
uint16_t	603
uint256	604
uint32_t	607
uint64_t	610
uint8_t	615
uint_error	616
valtype	616
valtype	617
valtype	617
valtype	617
zero_after_free_allocator	618
base	618
const_pointer.....	619
const_reference	619
difference_type.....	620
pointer	620
rebind	620
other	621
reference	621
size_type	621
value_type.....	622
Consensus	622
Params.....	623
leveldb	625
Arena.....	626
Block	628
BlockBuilder	630

BlockContents	632
BlockHandle	633
<anonymous>	635
Cache	636
Handle	638
Compaction	638
Comparator	642
CompressionType	643
DB	644
DBImpl	647
CompactionStats	660
ManualCompaction	661
Env	663
EnvWrapper	667
FileLock	670
FileMetaData	671
FileType	673
FilterBlockBuilder	675
FilterBlockReader	677
FilterPolicy	678
Footer	680
<anonymous>	681
Histogram	682
<anonymous>	684
Iterator	685
Cleanup	687
IteratorWrapper	688
Logger	691
MemTable	692
KeyComparator	695
Table	696
Node	696
Options	698
PosixLogger	702
Random	703
RandomAccessFile	704
Range	705
ReadOptions	707
SequentialFile	708
SkipList	709
<anonymous>	713
Iterator	713
Slice	715
Snapshot	718
SnapshotImpl	718
SnapshotList	720

Status	722
Code	724
Table	726
TableBuilder	728
TableCache	731
Version	734
GetStats	739
VersionEdit	740
DeletedFileSet	745
VersionSet	745
LevelSummaryStorage	753
WritableDatabase	754
WriteBatch	755
Handler	756
WriteBatchInternal	757
WriteOptions	759
log	760
Reader	761
<anonymous>	765
Reporter	765
RecordType	766
Writer	767
port	770
AtomicPointer	770
AtomicPointer	771
AtomicPointer	772
CondVar	774
CondVar	776
CondVar	777
Mutex	778
Mutex	779
Mutex	781
OnceType	782
OnceType	783
OnceType	783
test	784
ErrorEnv	784
memusage	786
boost_unordered_node	786
stl_tree_node	787
payments	788
Output	789
Payment	795
PaymentACK	804
PaymentDetails	809
PaymentRequest	820

X509Certificates.....	830
-----------------------	-----

Model Documentation

Model Detail

This document provides a complete overview of all element details. For simpler and more focused reports, simply copy this initial template and turn off the sections not required.

Class Model

Type: **Package**
Status: Proposed. Version . Phase 1.0.
Package: Model
Detail: *Created on 8/30/2015. Last modified on 8/30/2015*
GUID: {381A27FB-0D96-490c-B9D5-8EBFF71DDB0E}

Class Model - (Logical diagram)

Created By: dev on 8/30/2015
Last Modified: 8/30/2015
Version: 1.0. **Locked:** False
GUID: {7DCB3B33-DB05-410e-90D2-777ACAE68E0E}

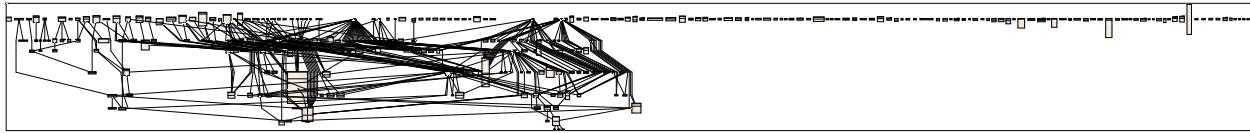


Figure: 1

<anonymous>

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {F8D502DB-F8BD-44a8-83C3-10F473837E1C}

Templates for serializing to anything that looks like a stream, i.e. anything that supports `.read(char*, size_t)` and `.write(char*, size_t)`

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
SER_NETWORK Public	primary actions	<i>Default:</i> (1 << 0)
SER_DISK Public		<i>Default:</i> (1 << 1)
SER_GETHASH Public		<i>Default:</i> (1 << 2)

<anonymous>

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {7776AE23-DFD5-4d49-A418-5604CBDB0DDF}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
MSG_TX Public		<i>Default:</i> 1

Attribute	Notes	Constraints and tags
MSG_BLOCK Public		<i>Default:</i>
MSG_FILTERED_BLOCK Public	Nodes may always request a MSG_FILTERED_BLOCK in a getdata, however, MSG_FILTERED_BLOCK should not appear in any invs except as a part of getdata.	<i>Default:</i>

<anonymous>

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {5ED41B32-09CE-4d3c-B13C-4B9A4E1C9860}

nServices flags

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
NODE_NETWORK Public	NODE_NETWORK means that the node is capable of serving the block chain. It is currently set by all Bitcoin Core nodes, and is unset by SPV clients or other peers that just want network services but don't provide them.	<i>Default:</i> (1 << 0)
NODE_GETUTXO Public	NODE_GETUTXO means the node is capable of responding to the getutxo protocol request. Bitcoin Core does not support this but a patch set called Bitcoin XT does. See BIP 64 for details on how this is implemented.	<i>Default:</i> (1 << 1)

<anonymous>**Type:****Enumeration****Status:**

Proposed. Version 1.0. Phase 1.0.

Package:

Class Model

Keywords:**Detail:**

Created on 8/30/2015. Last modified on 8/30/2015.

GUID:

{FF084F50-5873-4665-9C4E-18CC30FEB032}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
LOCAL_NONE Public	unknown	<i>Default:</i>
LOCAL_IF Public	address a local interface listens on	<i>Default:</i>
LOCAL_BIND Public	address explicit bound to	<i>Default:</i>
LOCAL_UPNP Public	address reported by UPnP	<i>Default:</i>

Attribute	Notes	Constraints and tags
LOCAL_MANUAL Public	address explicitly specified (-externalip=)	<i>Default:</i>
LOCAL_MAX Public		<i>Default:</i>

<anonymous>

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {BBB1706C-9508-4c2e-8FE4-DF214F72C528}

Script verification flags

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
SCRIPT_VERIFY_NONE Public		<i>Default:</i> 0
SCRIPT_VERIFY_P2SH Public	Evaluate P2SH subscripts (softfork safe, BIP16).	<i>Default:</i> (1U << 0)

Attribute	Notes	Constraints and tags
SCRIPT_VERIFY_STRICTENC Public	Passing a non-strict-DER signature or one with undefined hashtype to a checksig operation causes script failure. Evaluating a pubkey that is not (0x04 + 64 bytes) or (0x02 or 0x03 + 32 bytes) by checksig causes script failure. (softfork safe, but not used or intended as a consensus rule).	<i>Default:</i> (1U << 1)
SCRIPT_VERIFY_DERSIG Public	Passing a non-strict-DER signature to a checksig operation causes script failure (softfork safe, BIP62 rule 1)	<i>Default:</i> (1U << 2)
SCRIPT_VERIFY_LOW_S Public	Passing a non-strict-DER signature or one with S > order/2 to a checksig operation causes script failure (softfork safe, BIP62 rule 5).	<i>Default:</i> (1U << 3)
SCRIPT_VERIFY_NULLDUMMY Public	verify dummy stack item consumed by CHECKMULTISIG is of zero-length (softfork safe, BIP62 rule 7).	<i>Default:</i> (1U << 4)
SCRIPT_VERIFY_SIGPUSHONLY Public	Using a non-push operator in the scriptSig causes script failure (softfork safe, BIP62 rule 2).	<i>Default:</i> (1U << 5)
SCRIPT_VERIFY_MINIMALDATA Public	Require minimal encodings for all push operations (OP_0... OP_16, OP_1NEGATE where possible, direct pushes up to 75 bytes, OP_PUSHDATA up to 255 bytes, OP_PUSHDATA2 for anything larger). Evaluating any other push causes the script to fail (BIP62 rule 3). In addition, whenever a stack element is interpreted as a number, it must be of minimal length (BIP62 rule 4). (softfork safe)	<i>Default:</i> (1U << 6)

Attribute	Notes	Constraints and tags
SCRIPT_VERIFY_DISCOURAGE_UPGRADEABLE_NOPS Public	Discourage use of NOPs reserved for upgrades (NOP1-10) Provided so that nodes can avoid accepting or mining transactions containing executed NOP's whose meaning may change after a soft-fork, thus rendering the script invalid; with this flag set executing discouraged NOPs fails the script. This verification flag will never be a mandatory flag applied to scripts in a block. NOPs that are not executed, e.g. within an unexecuted IF ENDIF block, are *not* rejected.	<i>Default:</i> (1U << 7)
SCRIPT_VERIFY_CLEANSTACK Public	Require that only a single stack element remains after evaluation. This changes the success criterion from "At least one stack element must remain, and when interpreted as a boolean, it must be true" to "Exactly one stack element must remain, and when interpreted as a boolean, it must be true". (softfork safe, BIP62 rule 6) Note: CLEANSTACK should never be used without P2SH.	<i>Default:</i> (1U << 8)
SCRIPT_VERIFY_CHECKLOCKTIMEVERIFY Public	Verify CHECKLOCKTIMEVERIFY See BIP65 for details.	<i>Default:</i> (1U << 9)

<anonymous>

Type:

Enumeration

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {71D8B611-29C5-4421-A992-1F264BA32CFB}

Signature hash types/flags

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
SIGHASH_ALL Public		<i>Default:</i> 1
SIGHASH_NONE Public		<i>Default:</i> 2
SIGHASH_SINGLE Public		<i>Default:</i> 3
SIGHASH_ANYONECA NPAY Public		<i>Default:</i> 0x80

<anonymous>

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A96045F5-3CEA-49ad-A8D9-DC528CF7D773}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.

Attributes

Attribute	Notes	Constraints and tags
colorizeIcons bool Private Const	Colorize single-color icons	<i>Default:</i>
imagesOnButtons bool Private Const	Show images on push buttons	<i>Default:</i>
platformId char* Private Const		<i>Default:</i>
useExtraSpacing bool Private Const	Extra padding/spacing in transactionview	<i>Default:</i>

<anonymous>

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {76A83FEE-25FD-4c65-BACC-9611A2FF577A}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.

Attributes

Attribute	Notes	Constraints and tags
appName char*		<i>Default:</i>
Private Const		
iconColorHueShift int		<i>Default:</i>
Private Const		
iconColorSaturationReduction int		<i>Default:</i>
Private Const		
networkId char*		<i>Default:</i>
Private Const		
titleAddText char*		<i>Default:</i>
Private Const		

AcceptedConnection

Type:	Class
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model Keywords:
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>
GUID:	{5E9E8E25-C27F-4a53-95DB-F6E8559803C2}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
~AcceptedConnection() Public		
Pure close() void Public		
Pure peer_address_to_string() std::string Public		
Pure stream() std::ostream& Public		

BanReason

Type:	Enumeration
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model Keywords:
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>
GUID:	{5A4DB66D-0209-47aa-93A8-BAE01EA9B329}

Custom Properties

- isActive = False

Tagged Values

- typedef = true.
- typeSynonyms = BanReason.

Attributes

Attribute	Notes	Constraints and tags
BanReasonUnknown Public		<i>Default:</i> 0
BanReasonNodeMisbehaving Public		<i>Default:</i> 1
BanReasonManuallyAdded Public		<i>Default:</i> 2

BaseSignatureChecker

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model

Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {AAE451E1-F978-46e8-916B-03FFD366B918}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public TransactionSignatureChecker	Public BaseSignatureChecker	
Generalization Source -> Destination	Public DummySignatureChecker	Public BaseSignatureChecker	

Operations

Method	Notes	Parameters
~BaseSignatureChecker() Public		
CheckLockTime() bool Public		CScriptNum& [in] nLockTime
CheckSig() bool Public		std::vector<unsigned char>& [in] scriptSig std::vector<unsigned char>& [in] vchPubKey CScript& [in] scriptCode

BaseSignatureCreator**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {C1BC11E7-BC38-4600-9C4C-A905FAC283B4}

Virtual base class for signature creators.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public BaseSignatureCreator	Protected keystore CKeyStore	
Generalization Source -> Destination	Public TransactionSignatureCreator	Public BaseSignatureCreator	
Generalization Source -> Destination	Public DummySignatureCreator	Public BaseSignatureCreator	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
keystore CKeyStore*		<i>Default:</i>

Operations

Method	Notes	Parameters
BaseSignatureCreator() Public		CKeyStore* [in] keystoreIn
~BaseSignatureCreator() Public		
Const Pure Checker() BaseSignatureChecker& Public		
Pure CreateSig() bool Public		std::vector<unsigned char>& [in] vchSig CKeyID& [in] keyid CScript& [in] scriptCode
Const KeyStore() CKeyStore& Public		

BasicTestingSetup

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {EE30AB66-1D10-4952-8CB0-C7838A21E3E1}

Basic testing setup. This just configures logging and chain parameters.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public TestingSetup	Public BasicTestingSetup	

Connector	Source	Target	Notes

Operations

Method	Notes	Parameters
BasicTestingSetup() Public		CBaseChainParams::Network [in] network
~BasicTestingSetup() Public		

BitcoinAddressCheckValidator

Type: **Class** **QValidator**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 G UID: {95768A73-E1EC-4501-9BFE-8627D7394351}

Bitcoin address widget validator, checks for a valid bitcoin address.

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
BitcoinAddressCheckValidator() Public		QObject* [in] parent
validate() State Public		QString& [in] input int& [in] pos

BitcoinAddressEntryValidator

Type: **Class** **QValidator**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 G UID: {1372D8C1-44ED-4791-9F15-684D5D4FB752}

Base58 entry widget validator, checks for valid characters and removes some whitespace.

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
BitcoinAddressEntryValidator() Public	Base58 characters are: "123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz" This is:- All numbers except for '0' - All upper-case letters except for 'T' and 'O' - All lower-case letters except for 'l'	QObject* [in] parent
validate() State Public		QString& [in] input int& [in] pos

BitcoinUnit

Type: **Class** **Unit**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 GUID: {E9DECFE5-8D94-49e0-83A1-FC31DB37FC30}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public BitcoinUnit	Public Unit	

BitcoinUnits

Type: **Class** **QAbstractListModel**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 GUID: {D4E5BB19-0858-43e6-BB87-04E676EF3D54}

Bitcoin unit definitions. Encapsulates parsing and formatting and serves as list model for drop-down selection boxes.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
unitlist QList<BitcoinUnits::Unit> Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Static availableUnits() QList<Unit> Public		
BitcoinUnits() Public		QObject* [in] parent
data() QVariant Public		QModelIndex& [in] index int [in] role
Static decimals() int Public		int [in] unit
Static description() QString Public		int [in] unit
Static factor() qint64 Public		int [in] unit
Static format() QString Public		int [in] unit CAmount& [in] amount bool [in] plussign SeparatorStyle [in] separators
Static formatHtmlWithUnit() QString Public		int [in] unit CAmount& [in] amount

Method	Notes	Parameters
		bool [in] plussign SeparatorStyle [in] separators
Static formatWithUnit() QString Public	NOTE: Using formatWithUnit in an HTML context risks wrapping quantities at the thousands separator. More subtly, it also results in a standard space rather than a thin space, due to a bug in Qt's XML whitespace canonicalisation. Please take care to use formatHtmlWithUnit instead, when appropriate.	int [in] unit CAmount& [in] amount bool [in] plussign SeparatorStyle [in] separators
Static getAmountColumnTitle() QString Public		int [in] unit
Static maxMoney() CAmount Public		
Static name() QString Public		int [in] unit
Static parse() bool Public		int [in] unit QString& [in] value CAmount* [in] val_out
Static removeSpaces() QString Public		QString [in] text
rowCount() int Public		QModelIndex& [in] parent
Static valid() bool Public		int [in] unit

RoleIndex

Type: **Enumeration**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 G UID: {A6FC6A82-3DCA-49e4-B0A3-CC07CC060106}

! @name AbstractListModel implementation ! List model for unit drop-down selection box.
 @{

Custom Properties

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
UnitRole Public	Unit identifier	<i>Default:</i> Qt::UserRole

SeparatorStyle

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {3446E4BB-9E47-4c70-A6C7-9536C4DD6AB2}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
separatorNever Public		<i>Default:</i>
separatorStandard Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
separatorAlways Public		<i>Default:</i>

Unit

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {69812E9F-C820-45f0-B149-C673A67BCB9E}

Bitcoin units.

@note Source: <https://en.bitcoin.it/wiki/Units> . Please add only sensible ones

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public BitcoinUnit	Public Unit	

Attributes

Attribute	Notes	Constraints and tags
BTC Public		<i>Default:</i>
mBTC Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
uBTC Public		<i>Default:</i>

BlockHasher

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {59AB8C6B-AD33-4a6d-83BC-DCE38878EA36}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
operator() size_t Public		uint256& [in] hash

BlockMap

Type: Class boost::unordered_map<uint256, CBlockIndex*, BlockHasher>
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {C1A0B680-5D4C-4ead-9689-C0D399435F95}

Custom Properties

- isActive = False

BlockStatus

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {48661C7F-185D-4eb5-BB63-EC27D7989023}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
BLOCK_VALID_UNKN OWN Public	! Unused.	<i>Default:</i> 0
BLOCK_VALID_HEAD ER Public	! Parsed, version ok, hash satisfies claimed PoW, $1 \leq \text{vtx count} \leq \text{max}$, timestamp not in future	<i>Default:</i> 1
BLOCK_VALID_TREE Public	! All parent headers found, difficulty matches, timestamp \geq median previous, checkpoint. Implies all parents ! are also at least TREE.	<i>Default:</i> 2
BLOCK_VALID_TRAN SACTIONS Public	Only first tx is coinbase, $2 \leq \text{coinbase input script length} \leq 100$, transactions valid, no duplicate txids, sigops, size, merkle root. Implies all parents are at least TREE but not necessarily TRANSACTIONS. When all parent blocks also have TRANSACTIONS, CBlockIndex::nChainTx will be set.	<i>Default:</i> 3

Attribute	Notes	Constraints and tags
BLOCK_VALID_CHAIN Public	! Outputs do not overspend inputs, no double spends, coinbase output ok, no immature coinbase spends, BIP30. ! Implies all parents are also at least CHAIN.	<i>Default:</i> 4
BLOCK_VALID_SCRIPTS Public	! Scripts & signatures ok. Implies all parents are also at least SCRIPTS.	<i>Default:</i> 5
BLOCK_VALID_MASK Public	! All validity bits.	<i>Default:</i> BLOCK_VALID_HEADER BLOCK_VALID_TREE BLOCK_VALID_TRANSACTION S BLOCK_VALID_CHAIN BLOCK_VALID_SCRIPTS
BLOCK_HAVE_DATA Public	! full block available in blk*.dat	<i>Default:</i> 8
BLOCK_HAVE_UNDO Public	! undo data available in rev*.dat	<i>Default:</i> 16
BLOCK_HAVE_MASK Public		<i>Default:</i> BLOCK_HAVE_DATA BLOCK_HAVE_UNDO

Attribute	Notes	Constraints and tags
BLOCK_FAILED_VALID Public	! stage after last reached validness failed	<i>Default:</i> 32
BLOCK_FAILED_CHILD Public	! descends from failed block	<i>Default:</i> 64
BLOCK_FAILED_MAX Public		<i>Default:</i> BLOCK_FAILED_VALID BLOCK_FAILED_CHILD

CAccount

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {3F051081-A744-423c-B9EF-2F5B2D5F2821}

Account information. Stored in wallet with key "acc"+string account name.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CAccount	Public vchPubKey CPubKey	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
vchPubKey CPubKey Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CAccount() Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		

CAccountingEntry**Type:** [Class](#)**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {1DBD4169-83EF-403c-B61F-72998B4777DE}

Internal transfers. Database key is acentry<account><counter>.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CAccountingEntry	Public mapValue mapValue_t	
Association Source -> Destination	Public CAccountingEntry	Public nCreditDebit CAmount	
Association Source -> Destination	Public CAccountingEntry	Public nEntryNo uint64_t	
Association Source -> Destination	Public CAccountingEntry	Public nOrderPos int64_t	
Association Source -> Destination	Public CAccountingEntry	Public nTime int64_t	

Attributes

Attribute	Notes	Constraints and tags
_ssExtra std::vector<char> Private		<i>Default:</i>
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
mapValue mapValue_t Public		<i>Default:</i>
nCreditDebit CAmount Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nEntryNo uint64_t Public		<i>Default:</i>
nOrderPos int64_t Public	! position in ordered transaction list	<i>Default:</i>
nTime int64_t Public		<i>Default:</i>
strAccount std::string Public		<i>Default:</i>
strComment std::string Public		<i>Default:</i>
strOtherAccount std::string Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CAccountingEntry()		

Method	Notes	Parameters
Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		

CAddrDB

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {C6D16E97-3277-40a3-B1B8-C6B82940A6E8}

Access to the (IP) address database (peers.dat)

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
pathAddr boost::filesystem::path Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CAddrDB() Public	CAddrDB	
Read() bool Public		CAddrMan& [in] addr
Write() bool Public		CAddrMan& [in] addr

CAddrInfo

Type: Class **CAddress**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D74BD039-3AA7-4f00-9DD8-181FEAA1BBD6}

Extended statistics about a CAddress

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CAddrInfo	Private nLastSuccess int64_t	
Association Source -> Destination	Public CAddrInfo	Public nLastTry int64_t	
Association Source -> Destination	Public CAddrInfo	Private source CNetAddr	
Generalization Source -> Destination	Public CAddrInfo	Public CAddress	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
fInTried bool Private	! in tried set? (memory only)	<i>Default:</i>

Attribute	Notes	Constraints and tags
nAttempts int Private	! connection attempts since last successful attempt	<i>Default:</i>
nLastSuccess int64_t Private	! last successful connection by us	<i>Default:</i>
nLastTry int64_t Public	! last try whatsoever by us (memory only)	<i>Default:</i>
nRandomPos int Private	! position in vRandom	<i>Default:</i>
nRefCount int Private	! reference count in new sets (memory only)	<i>Default:</i>
source CNetAddr Private	! where knowledge about this address first came from	<i>Default:</i>

Operations

Method	Notes	Parameters
CAddrInfo()		CAddress& [in] addrIn

Method	Notes	Parameters
Public		CNetAddr& [in] addrSource
CAddrInfo() Public		
GetBucketPosition() int Public		uint256& [in] nKey bool [in] fNew int [in] nBucket
GetChance() double Public		int64_t [in] nNow
GetNewBucket() int Public		uint256& [in] nKey CNetAddr& [in] src
GetNewBucket() int Public		uint256& [in] nKey
GetTriedBucket() int Public		uint256& [in] nKey
Init() void Public		
IsTerrible() bool Public		int64_t [in] nNow
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion

CAddrMan

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {D2B3F497-BD56-437b-8401-6A03F2C3C2CF}

Stochastical (IP) address manager

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CAddrMan	Private nKey uint256	

Attributes

Attribute	Notes	Constraints and tags
cs CCriticalSection Private	! critical section to protect the inner data structures	<i>Default:</i> [<u>mutable</u> = true]
mapAddr std::map<CNetAddr, int> Private	! find an nId based on its network address	<i>Default:</i>
mapInfo std::map<int, CAddrInfo> Private	! table with information about all nIds	<i>Default:</i>
nIdCount int Private	! last used nId	<i>Default:</i>

Attribute	Notes	Constraints and tags
nKey uint256 Private	! secret key to randomize bucket select with	<i>Default:</i>
nNew int Private	! number of (unique) "new" entries	<i>Default:</i>
nTried int Private	number of "tried" entries	<i>Default:</i>
vRandom std::vector<int> Private	! randomly-ordered vector of all nIds	<i>Default:</i>
vvNew int Private Collection	! list of "new" buckets	<i>Default:</i>
vvTried int Private Collection	! list of "tried" buckets	<i>Default:</i>

Operations

Method	Notes	Parameters
Add() bool		CAddress& [in] addr

Method	Notes	Parameters
Public		CNetAddr& [in] source int64_t [in] nTimePenalty
Add() bool Public		std::vector<CAddress>& [in] vAddr CNetAddr& [in] source int64_t [in] nTimePenalty
Add_() bool Protected		CAddress& [in] addr CNetAddr& [in] source int64_t [in] nTimePenalty
Attempt() void Public		CService& [in] addr int64_t [in] nTime
Attempt_() void Protected		CService& [in] addr int64_t [in] nTime
CAddrMan() Public		
~CAddrMan() Public		
Check() void Public		
Check_() int Protected		
Clear() void Public		
ClearNew() void Protected		int [in] nUBucket int [in] nUBucketPos
Connected() void Public		CService& [in] addr int64_t [in] nTime
Connected_() void Protected		CService& [in] addr int64_t [in] nTime

Method	Notes	Parameters
Create() CAddrInfo*		
Protected		CAddress& [in] addr CNetAddr& [in] addrSource int* [in] pnId
Delete() void		int [in] nId
Protected		
Find() CAddrInfo*		
Protected		CNetAddr& [in] addr int* [in] pnId
GetAddr()		
std::vector<CAddress>		
Public		
GetAddr_() void		
Protected		std::vector<CAddress>& [in] vAddr
GetSerializeSize()		
unsigned int		int [in] nType
Public		int [in] nVersion
Good() void		
Public		CService& [in] addr int64_t [in] nTime
Good_() void		
Protected		CService& [in] addr int64_t [in] nTime
MakeTried() void		
Protected		CAddrInfo& [in] info int [in] nId
Select() CAddrInfo		
Public		
Select_() CAddrInfo		
Protected		
Serialize() void		
Public		Stream& [in] s int [in] nType int [in] nVersionDummy
size() size_t		

Method	Notes	Parameters
Public		
SwapRandom() void Protected		unsigned int [in] nRandomPos1 unsigned int [in] nRandomPos2
Unserialize() void Public		Stream& [in] s int [in] nType int [in] nVersionDummy

CAddress

Type: [Class](#) [CService](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {DF365E92-4CBB-4cb9-BA23-D5213954F3D3}

A CService with information about it as peer

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CNode	Public addr CAddress	
Association Source -> Destination	Public CAddress	Public nServices uint64_t	
Generalization Source -> Destination	Public CAddress	Public CService	
Generalization Source -> Destination	Public CAddrInfo	Public CAddress	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
nServices uint64_t Public		<i>Default:</i>
nTime unsigned int Public	disk and network only	<i>Default:</i>

Operations

Method	Notes	Parameters
CAddress() Public		
CAddress() Public		CService [in] ipIn uint64_t [in] nServicesIn
Init() void Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion

CAddressBookData**Type:****Status:****Package:****Class**

Proposed. Version 1.0. Phase 1.0.

Class Model **Keywords:**

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {6AEA55E4-6310-4c8a-8A8F-071CA813F864}

Address book data

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CAddressBookData	Public destdata StringMap	

Attributes

Attribute	Notes	Constraints and tags
destdata StringMap Public		<i>Default:</i>
name std::string Public		<i>Default:</i>
purpose std::string Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CAddressBookData()		

StringMap

Type: **Class** **`std::map<std::string, std::string>`**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {6EFFD98C-B4AD-4417-B1D7-C22BB453E238}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CAddressBookData	Public destdata StringMap	

CAffectedKeysVisitor

Type: **Class** **`boost::static_visitor`**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D9DB44EB-5D6A-4891-A222-DC5C59A2316A}

end of Actions

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CAffectedKeysVisitor	Private keystore CKeyStore	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
keystore CKeyStore& Private Const		<i>Default:</i>
vKeys std::vector<CKeyID>& Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CAffectedKeysVisitor() Public		CKeyStore& [in] keystoreIn std::vector<CKeyID>& [in] vKeysIn
operator()() void Public		CKeyID& [in] keyId
operator()() void Public		CScriptID& [in] scriptId
operator()() void Public		CNoDestination& [in] none
Process() void Public		CScript& [in] script

CAlert

Type: **Class** **CUnsignedAlert**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A5AEE409-02E9-4184-9FF9-50E6FEDD2050}

An alert is a combination of a serialized CUnsignedAlert and a signature.

Custom Properties

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CAAlert	Public CUnsignedAlert	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
vchMsg std::vector<unsigned char> Public		<i>Default:</i>
vchSig std::vector<unsigned char> Public		<i>Default:</i>

Operations

Method	Notes	Parameters
AppliesTo() bool Public		int [in] nVersion std::string& [in] strSubVerIn
AppliesToMe() bool Public		
CAAlert() Public		
Cancels() bool Public		CAAlert& [in] alert

Method	Notes	Parameters
CheckSignature() bool Public		std::vector<unsigned char>& [in] alertKey
Static getAlertByHash() CAler Public		uint256& [in] hash
GetHash() uint256 Public		
IsInEffect() bool Public		
IsNull() bool Public		
Static Notify() void Public		std::string& [in] strMessage bool [in] fThread
ProcessAlert() bool Public		std::vector<unsigned char>& [in] alertKey bool [in] fThread
RelayTo() bool Public		CNode* [in] pnode
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		

CAmount

Type: **Class** **int64_t**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 GUID: {AC3ED576-7561-4de3-8B0B-CED3A7618CEF}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TransactionRecord	Public credit CAmount	
Association Source -> Destination	Public TransactionFilterProxy	Private minAmount CAmount	
Association Source -> Destination	Public TransactionRecord	Public debit CAmount	
Association Source -> Destination	Public CTxOut	Public nValue CAmount	
Association Source -> Destination	Public WalletModelTransactio n	Private fee CAmount	
Association Source -> Destination	Public CTxMemPoolEntry	Private nFee CAmount	
Association Source -> Destination	Public CAccountingEntry	Public nCreditDebit CAmount	
Generalization Source -> Destination	Public CAmount	Public int64_t	
Association Source -> Destination	Public CFeeRate	Private nSatoshisPerK CAmount	
Association Source -> Destination	Public CWalletTx	Public nChangeCached CAmount	
Association Source -> Destination	Public CRecipient	Public nAmount CAmount	
Association Source -> Destination	Public COutputEntry	Public amount CAmount	
Association Source -> Destination	Public CWalletTx	Public nAvailableWatchCredit Cached CAmount	
Association Source -> Destination	Public CWalletTx	Public nCreditCached CAmount	
Association Source -> Destination	Public CWalletTx	Public nDebitCached CAmount	
Association Source -> Destination	Public CWalletTx	Public nImmatureCreditCache	

Connector	Source	Target	Notes
		d CAmount	
Association Source -> Destination	Public CWalletTx	Public nImmatureWatchCredit Cached CAmount	
Association Source -> Destination	Public CWalletTx	Public nAvailableCreditCache d CAmount	
Association Source -> Destination	Public CCoinsStats	Public nTotalAmount CAmount	
Association Source -> Destination	Public CWalletTx	Public nWatchCreditCached CAmount	
Association Source -> Destination	Public CWalletTx	Public nWatchDebitCached CAmount	

CAutoFile

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {FD462E7C-0193-4a21-9462-C5D031538F31}

Non-refcounted RAII wrapper for FILE*. Will automatically close the file when it goes out of scope if not null. If you're returning the file pointer, return file.release(). If you need to close the file early, use file.fclose() instead of fclose(file).

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
file FILE* Private		<i>Default:</i>
nType int Private		<i>Default:</i>
nVersion int Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CAutoFile() Private		CAutoFile& [in] prm1
CAutoFile() Public		FILE* [in] filenew int [in] nTypeIn int [in] nVersionIn
~CAutoFile() Public		
fclose() void Public		
Get() FILE* Public		
GetSerializeSize() unsigned int Public		T& [in] obj
GetType() int Public		
GetVersion() int Public		
IsNull() bool Public		

Method	Notes	Parameters
operator<<() CAutoFile& Public		T& [in] obj
operator=() CAutoFile& Private		CAutoFile& [in] prm1
operator>>() CAutoFile& Public		T& [in] obj
read() CAutoFile& Public		char* [in] pch size_t [in] nSize
ReadVersion() void Public		
release() FILE* Public		
SetType() void Public		int [in] n
SetVersion() void Public		int [in] n
write() CAutoFile& Public		char* [in] pch size_t [in] nSize
WriteVersion() void Public		

CBanDB

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {8A2BAE42-41CE-4560-9B04-363ED6D77E08}

Access to the banlist database (banlist.dat)

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
pathBanlist boost::filesystem::path Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CBanDB() Public	CBanDB	
Read() bool Public		banmap_t& [in] banSet
Write() bool Public		banmap_t& [in] banSet

CBanEntry**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {C48A44DF-D360-45b9-9F4E-5009A4E0ADFB}**Custom Properties**

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBanEntry	Public nCreateTime int64_t	
Association Source -> Destination	Public CBanEntry	Public banReason uint8_t	
Association Source -> Destination	Public CBanEntry	Public nBanUntil int64_t	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
banReason uint8_t Public		<i>Default:</i>
CURRENT_VERSION int Public Static Const		<i>Default:</i> 1
nBanUntil int64_t Public		<i>Default:</i>
nCreateTime int64_t Public		<i>Default:</i>
nVersion int Public		<i>Default:</i>

Operations

Method	Notes	Parameters
banReasonToString()		

Method	Notes	Parameters
std::string Public CBanEntry() Public		
CBanEntry() Public		int64_t [in] nCreateTimeIn
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		

CBase58Data

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {0C868549-2398-46ec-BCF6-1D05CD80E44C}

Base class for all base58-encoded data

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBitcoinExtKeyBase	Public CBase58Data	
Generalization Source -> Destination	Public CBitcoinSecret	Public CBase58Data	
Association Source -> Destination	Public CBase58Data	Protected vchData vector_uchar	
Generalization Source -> Destination	Public CBitcoinAddress	Public CBase58Data	

Attributes

Attribute	Notes	Constraints and tags
vchData vector_uchar Protected		<i>Default:</i>
vchVersion std::vector<unsigned char> Protected	! the version byte(s)	<i>Default:</i>

Operations

Method	Notes	Parameters
CBase58Data() Protected		
CompareTo() int Public		CBase58Data& [in] b58
operator<() bool Public		CBase58Data& [in] b58
operator<=() bool Public		CBase58Data& [in] b58
operator==() bool Public		CBase58Data& [in] b58
operator>() bool Public		CBase58Data& [in] b58
operator>=() bool Public		CBase58Data& [in] b58
SetData() void Protected		std::vector<unsigned char>& [in] vchVersionIn void* [in] pdata size_t [in] nSize
SetData() void Protected		std::vector<unsigned char>& [in] vchVersionIn unsigned char* [in] pbegin unsigned char* [in] pend

Method	Notes	Parameters
SetString() bool Public		char* [in] psz unsigned int [in] nVersionBytes
SetString() bool Public		std::string& [in] str
ToString() std::string Public		

vector_uchar

Type: **Class** [std::vector<unsigned char, zero_after_free_allocator<unsigned char> >](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model [Keywords:](#)

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {1CC51389-39D9-4698-AEE3-7F444A2CF968}

! the actually encoded data

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBase58Data	Protected vchData vector_uchar	

CBaseChainParams

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model [Keywords:](#)

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {ED1C1131-C0B4-4b9d-8435-2A6E0049674D}

CBaseChainParams defines the base parameters (shared between bitcoin-cli and bitcoind) of a given instance of the Bitcoin system.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBaseMainParams	Public CBaseChainParams	
Generalization Source -> Destination	Public CBaseTestNetParams	Public CBaseChainParams	
Generalization Source -> Destination	Public CBaseRegTestParams	Public CBaseChainParams	

Attributes

Attribute	Notes	Constraints and tags
nRPCPort int Protected		<i>Default:</i>
strDataDir std::string Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
CBaseChainParams() Protected		
Const DataDir() std::string& Public		
RPCPort() int Public		

Network

Type:**Enumeration****Status:**

Proposed. Version 1.0. Phase 1.0.

Package:

Class Model

Keywords:**Detail:**

Created on 8/30/2015. Last modified on 8/30/2015.

GUID:

{62684C7D-CBD2-41b8-A6E5-05EA4688012B}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
MAIN Public		<i>Default:</i>
TESTNET Public		<i>Default:</i>
REGTEST Public		<i>Default:</i>
MAX_NETWORK_TYPES Public		<i>Default:</i>

CBaseMainParams

Type: **Class** [**CBaseChainParams**](#)
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model *Keywords:*
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 GUID: {7AA5DE04-9F83-4bbc-B16E-7C90D6619A23}

Main network

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBaseMainParams	Public CBaseChainParams	
Generalization Source -> Destination	Public CBaseUnitTestParams	Public CBaseMainParams	

Operations

Method	Notes	Parameters
CBaseMainParams() Public		

CBaseRegTestParams

Type: **Class** **CBaseChainParams**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {9DA1EF89-FC75-4ca5-BD5C-815F97C0C37E}

Regression test

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBaseRegTestParams	Public CBaseChainParams	

Operations

Method	Notes	Parameters
CBaseRegTestParams() Public		

CBaseTestNetParams

Type:	Class	CBaseChainParams
Status:	Proposed.	Version 1.0. Phase 1.0.
Package:	Class Model	<i>Keywords:</i>
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>	
GUID:	{022D4A55-7CC0-44c0-A969-29F1F8DB2004}	

Testnet (v3)

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBaseTestNetParams	Public CBaseChainParams	

Operations

Method	Notes	Parameters
CBaseTestNetParams() Public		

CBaseUnitTestParams

Type:	Class	CBaseMainParams
Status:	Proposed.	Version 1.0. Phase 1.0.
Package:	Class Model	<i>Keywords:</i>
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>	
GUID:	{E2F6A0AD-BFB7-429b-B21C-920CA8C1080F}	

Unit test

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBaseUnitTestParams	Public CBaseMainParams	

Operations

Method	Notes	Parameters
--------	-------	------------

Method	Notes	Parameters
CBaseUnitTestParams() Public		

CBasicKeyStore

Type: Class **CKeyStore**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {E4F4B544-39DA-4efa-A2C6-CCF6209E627E}

Basic key store, that keeps keys in an address->secret map

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBasicKeyStore	Public CKeyStore	
Association Source -> Destination	Public CBasicKeyStore	Protected mapKeys KeyMap	
Association Source -> Destination	Public CBasicKeyStore	Protected mapScripts ScriptMap	
Association Source -> Destination	Public CBasicKeyStore	Protected setWatchOnly WatchOnlySet	
Association Source -> Destination	Public CBasicKeyStore	Protected mapWatchKeys WatchKeyMap	
Generalization Source -> Destination	Public CCryptoKeyStore	Public CBasicKeyStore	

Attributes

Attribute	Notes	Constraints and tags
mapKeys KeyMap Protected		<i>Default:</i>

Attribute	Notes	Constraints and tags
mapScripts ScriptMap Protected		<i>Default:</i>
mapWatchKeys WatchKeyMap Protected		<i>Default:</i>
setWatchOnly WatchOnlySet Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
AddCScript() bool Public		CScript& [in] redeemScript
AddKeyPubKey() bool Public		CKey& [in] key CPubKey& [in] pubkey
AddWatchOnly() bool Public		CScript& [in] dest
GetCScript() bool Public		CScriptID& [in] hash CScript& [in] redeemScriptOut
GetKey() bool Public		CKeyID& [in] address CKey& [in] keyOut
GetKeys() void Public		std::set<CKeyID>& [in] setAddress

Method	Notes	Parameters
GetPubKey() bool Public		CKeyID& [in] address CPubKey& [in] vchPubKeyOut
HaveCScript() bool Public		CScriptID& [in] hash
HasKey() bool Public		CKeyID& [in] address
HaveWatchOnly() bool Public		CScript& [in] dest
HaveWatchOnly() bool Public		
RemoveWatchOnly() bool Public		CScript& [in] dest

CBigNum

Type: **Class** **BIGNUM**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 GUID: {2A53FC9E-D789-42f4-AD45-CAF3B14F71A3}

C++ wrapper for BIGNUM (OpenSSL bignum)

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
CBigNum() Public		
CBigNum() Public		CBigNum& [in] b
~CBigNum() Public		
CBigNum() Public		long long [in] n
CBigNum() Public		std::vector<unsigned char>& [in] vch

Method	Notes	Parameters
getint() int Public		
getvch() std::vector<unsigned char> Public		
Const operator-() CBigNum Public		CBigNum& [in] a CBigNum& [in] b
operator=() CBigNum& Public		CBigNum& [in] b
setint64() void Public		int64_t [in] sn
setvch() void Public		std::vector<unsigned char>& [in] vch

CBitcoinAddress

Type: **Class** [CBase58Data](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model [Keywords](#):

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {DACC3BB9-54B3-439d-BE7B-391F408D5E47}

base58-encoded Bitcoin addresses. Public-key-hash-addresses have version 0 (or 111 testnet). The data vector contains RIPEMD160(SHA256(pubkey)), where pubkey is the serialized public key. Script-hash-addresses have version 5 (or 196 testnet). The data vector contains RIPEMD160(SHA256(cscript)), where cscript is the serialized redemption script.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBitcoinAddressVisitor	Private addr CBitcoinAddress	
Generalization Source -> Destination	Public CBitcoinAddress	Public CBase58Data	

Operations

Method	Notes	Parameters
--------	-------	------------

Method	Notes	Parameters
CBitcoinAddress() Public		
CBitcoinAddress() Public		CTxDestination& [in] dest
CBitcoinAddress() Public		std::string& [in] strAddress
CBitcoinAddress() Public		char* [in] pszAddress
Get() CTxDestination Public		
GetKeyID() bool Public		CKeyID& [in] keyID
IsScript() bool Public		
IsValid() bool Public		
IsValid() bool Public		CChainParams& [in] params
Set() bool Public	anon namespace	CKeyID& [in] id
Set() bool Public		CScriptID& [in] id
Set() bool Public		CTxDestination& [in] dest

CBitcoinAddressVisitor

Type: Class **boost::static_visitor**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {369E5963-A27F-4ec1-AE71-CD2C6530C6A3}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association	Public	Private addr	

Connector	Source	Target	Notes
Source -> Destination	CBitcoinAddressVisitor	CBitcoinAddress	

Attributes

Attribute	Notes	Constraints and tags
addr CBitcoinAddress*	Private	<i>Default:</i>

Operations

Method	Notes	Parameters
CBitcoinAddressVisitor() Public		CBitcoinAddress* [in] addrIn
operator() bool Public		CKeyID& [in] id
operator() bool Public		CScriptID& [in] id
operator() bool Public		CNoDestination& [in] no

CBitcoinExtKey

Type: Class [CBitcoinExtKeyBase<CExtKey, 74, CChainParams::EXT_SECRET_KEY>](#)
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model [Keywords](#):
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 G UID: {2E0272BF-8BB1-415b-9514-163AD33275A4}

Custom Properties

- isActive = False

CBitcoinExtKeyBase

Type: Class [CBase58Data](#)
 Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {1F0DC4EC-A4B0-4b6b-8484-35CEE5594633}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBitcoinExtKeyBase	Public CBase58Data	

Operations

Method	Notes	Parameters
CBitcoinExtKeyBase() Public		K& [in] key
CBitcoinExtKeyBase() Public		std::string& [in] strBase58c
CBitcoinExtKeyBase() Public		
GetKey() K Public		
SetKey() void Public		K& [in] key

CBitcoinExtPubKey

Type: Class **CBitcoinExtKeyBase<CExtPubKey, 74, CChainParams::EXT_PUBLIC_KEY>**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {F5EFC2A5-1FC0-4750-885B-FFC8D14DD9A2}

Custom Properties

- isActive = False

CBitcoinSecret

Type: **Class** [CBase58Data](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {C54CFF4A-AEFE-42db-8A18-D0E07D6AD60A}

A base58-encoded secret key

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBitcoinSecret	Public CBase58Data	

Operations

Method	Notes	Parameters
CBitcoinSecret() Public		CKey& [in] vchSecret
CBitcoinSecret() Public		
GetKey() CKey Public		
IsValid() bool Public		
SetKey() void Public		CKey& [in] vchSecret
SetString() bool Public		char* [in] pszSecret
SetString() bool Public		std::string& [in] strSecret

CBlock

Type: **Class** [CBlockHeader](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {0867808E-E7F0-471c-87C2-9B35E22CC772}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBlock	Public CBlockHeader	
Association Source -> Destination	Public CBlockTemplate	Public block CBlock	
Association Source -> Destination	Public CChainParams	Protected genesis CBlock	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
vMerkleTree std::vector<uint256> Public	memory only	<i>Default:</i> [<u>mutable</u> = true]
vtx std::vector<CTransaction> Public	network and disk	<i>Default:</i>

Operations

Method	Notes	Parameters
BuildMerkleTree() uint256 Public		bool* [in] mutated
CBlock() Public		

Method	Notes	Parameters
CBlock() Public		CBlockHeader& [in] header
Static CheckMerkleBranch() uint256 Public		uint256 [in] hash std::vector<uint256>& [in] vMerkleBranch int [in] nIndex
GetBlockHeader() CBlockHeader Public		
GetMerkleBranch() std::vector<uint256> Public		int [in] nIndex
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		
ToString() std::string Public		

CBlockFileInfo

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {F42D8527-0C5B-4176-850F-30EA18E2603C}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBlockFileInfo	Public nTimeFirst uint64_t	

Connector	Source	Target	Notes
Association Source -> Destination	Public CBlockFileInfo	Public nTimeLast uint64_t	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
nBlocks unsigned int Public	! number of blocks stored in file	<i>Default:</i>
nHeightFirst unsigned int Public	! lowest height of block in file	<i>Default:</i>
nHeightLast unsigned int Public	! highest height of block in file	<i>Default:</i>
nSize unsigned int Public	! number of used bytes of block file	<i>Default:</i>

Attribute	Notes	Constraints and tags
nTimeFirst uint64_t Public	! earliest time of block in file	<i>Default:</i>
nTimeLast uint64_t Public	! latest time of block in file	<i>Default:</i>
nUndoSize unsigned int Public	! number of used bytes in the undo file	<i>Default:</i>

Operations

Method	Notes	Parameters
AddBlock() void Public		unsigned int [in] nHeightIn uint64_t [in] nTimeIn
CBlockFileInfo() Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		
ToString() std::string Public		

CBlockHeaderType: [Class](#)

Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {AE530AC1-BED2-49ff-AFB8-56F0E523B17D}

Nodes collect new transactions into a block, hash them into a hash tree, and scan through nonce values to make the block's hash satisfy proof-of-work requirements. When they solve the proof-of-work, they broadcast the block to everyone and the block is added to the block chain. The first transaction in the block is a special one that creates a new coin owned by the creator of the block.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBlockHeader	Public hashMerkleRoot uint256	
Association Source -> Destination	Public CBlockHeader	Public hashPrevBlock uint256	
Association Source -> Destination	Public CBlockHeader	Public nBits uint32_t	
Association Source -> Destination	Public CBlockHeader	Public nNonce uint32_t	
Association Source -> Destination	Public CBlockHeader	Public nTime uint32_t	
Association Source -> Destination	Public CBlockHeader	Public nVersion int32_t	
Association Source -> Destination	Public CBlockHeader	Public CURRENT_VERSION int32_t	
Generalization Source -> Destination	Public CBlock	Public CBlockHeader	
Association Source -> Destination	Public CMerkleBlock	Public header CBlockHeader	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
CURRENT_VERSION int32_t Public Static Const	header	<i>Default:</i> 3
hashMerkleRoot uint256 Public		<i>Default:</i>
hashPrevBlock uint256 Public		<i>Default:</i>
nBits uint32_t Public		<i>Default:</i>
nNonce uint32_t Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nTime uint32_t Public		<i>Default:</i>
nVersion int32_t Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CBlockHeader() Public		
GetBlockTime() int64_t Public		
GetHash() uint256 Public		
IsNull() bool Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		

CBlockIndex

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {F85697C0-98AE-4646-AD00-792C57CECF31}

The block chain is a tree shaped structure starting with the genesis block at the root, with each block potentially having multiple candidates to be the next block. A blockindex may have multiple pprev pointing to it, but at most one of them can be part of the currently active branch.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CDiskBlockIndex	Public CBlockIndex	
Association Source -> Destination	Public CBlockIndex	Public hashMerkleRoot uint256	
Association Source -> Destination	Public CBlockIndex	Public nChainWork arith_uint256	
Association Source -> Destination	Public CBlockIndex	Public nSequenceId uint32_t	
Association Source -> Destination	Public CBlockIndex	Public phashBlock uint256	
Association Source -> Destination	Public CBlockIndex	Public pprev CBlockIndex	
Association Source -> Destination	Public CBlockIndex	Public pskip CBlockIndex	

Attributes

Attribute	Notes	Constraints and tags
hashMerkleRoot uint256 Public		<i>Default:</i>
nBits unsigned int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nChainTx unsigned int Public	! (memory only) Number of transactions in the chain up to and including this block. ! This value will be non-zero only if and only if transactions for this block and all its parents are available. ! Change to 64-bit type when necessary; won't happen before 2030	<i>Default:</i>
nChainWork arith_uint256 Public	! (memory only) Total amount of work (expected number of hashes) in the chain up to and including this block	<i>Default:</i>
nDataPos unsigned int Public	! Byte offset within blk?????.dat where this block's data is stored	<i>Default:</i>
nFile int Public	! Which # file this block is stored in (blk?????.dat)	<i>Default:</i>
nHeight int Public	! height of the entry in the chain. The genesis block has height 0	<i>Default:</i>
nNonce unsigned int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nSequenceId uint32_t Public	! (memory only) Sequential id assigned to distinguish order in which blocks are received.	<i>Default:</i>
nStatus unsigned int Public	! Verification status of this block. See enum BlockStatus	<i>Default:</i>
nTime unsigned int Public		<i>Default:</i>
nTx unsigned int Public	! Number of transactions in this block. ! Note: in a potential headers-first mode, this number cannot be relied upon	<i>Default:</i>
nUndoPos unsigned int Public	! Byte offset within rev?????.dat where this block's undo data is stored	<i>Default:</i>
nVersion int Public	! block header	<i>Default:</i>

Attribute	Notes	Constraints and tags
phashBlock uint256*	! pointer to the hash of the block, if any. Memory is owned by this CBlockIndex Public Const	<i>Default:</i>
pprev CBlockIndex*	! pointer to the index of the predecessor of this block Public	<i>Default:</i>
pskip CBlockIndex*	! pointer to the index of some further predecessor of this block Public	<i>Default:</i>

Operations

Method	Notes	Parameters
BuildSkip() void		
Public		
CBlockIndex()		
Public		
CBlockIndex()		CBlockHeader& [in] block
Public		
GetAncestor()		int [in] height
CBlockIndex*		
Public		
GetAncestor()		int [in] height
CBlockIndex*		
Public		
GetBlockHash() uint256		
Public		
GetBlockHeader()		
CBlockHeader		
Public		
GetBlockPos()		
CDiskBlockPos		
Public		
GetBlockTime() int64_t		
Public		
GetMedianTimePast()		
int64_t		

Method	Notes	Parameters
Public		
GetUndoPos() CDiskBlockPos Public		
IsValid() bool Public		enum BlockStatus [in] nUpTo
RaiseValidity() bool Public		enum BlockStatus [in] nUpTo
SetNull() void Public		
ToString() std::string Public		

<anonymous>

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A607AED0-A140-449f-9230-32CD4E3A103B}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
nMedianTimeSpan Public		<i>Default: 11</i>

CBlockLocator

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {E6DE56C3-5FE1-429c-87B8-3B96ED0C2F69}

Describes a place in the block chain to another node such that if the other node doesn't have the same branch, it can find a recent common trunk. The further back it is, the further before the fork it may be.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
vHave std::vector<uint256> Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CBlockLocator() Public		
CBlockLocator() Public		std::vector<uint256>& [in] vHaveIn
IsNull() bool Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		

CBlockPolicyEstimator

Type:	Class
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model
Detail:	Created on 8/30/2015. Last modified on 8/30/2015.
GUID:	{2A6C1AA4-9925-4fbf-89A7-DB8844282E64}

We want to be able to estimate fees or priorities that are needed on tx's to be included in a certain number of blocks. Every time a block is added to the best chain, this class records stats on the transactions included in that block

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBlockPolicyEstimator	Private feeUnlikely CFeeRate	
Association Source -> Destination	Public CBlockPolicyEstimator	Private feeStats TxConfirmStats	
Association Source -> Destination	Public CBlockPolicyEstimator	Private minTrackedFee CFeeRate	
Association Source -> Destination	Public CBlockPolicyEstimator	Private priStats TxConfirmStats	
Association Source -> Destination	Public CBlockPolicyEstimator	Private feeLikely CFeeRate	
Association Source -> Destination	Public CTxMemPool	Private minerPolicyEstimator CBlockPolicyEstimator	

Attributes

Attribute	Notes	Constraints and tags
feeLikely CFeeRate Private	Breakpoints to help determine whether a transaction was confirmed by priority or Fee	<i>Default:</i>

Attribute	Notes	Constraints and tags
feeStats TxConfirmStats Private	Classes to track historical data on transaction confirmations	<i>Default:</i>
feeUnlikely CFeeRate Private	Breakpoints to help determine whether a transaction was confirmed by priority or Fee	<i>Default:</i>
mapMemPoolTxs std::map<uint256, TxStatsInfo> Private	map of txids to information about that transaction	<i>Default:</i>
minTrackedFee CFeeRate Private	! Passed to constructor to avoid dependency on main	<i>Default:</i>
minTrackedPriority double Private	! Set to AllowFreeThreshold	<i>Default:</i>
nBestSeenHeight unsigned int Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
priLikely double Private		<i>Default:</i>
priStats TxConfirmStats Private	Classes to track historical data on transaction confirmations	<i>Default:</i>
priUnlikely double Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CBlockPolicyEstimator() Public		CFeeRate& [in] minRelayFee
estimateFee() CFeeRate Public		int [in] confTarget
estimatePriority() double Public		int [in] confTarget
isFeeDataPoint() bool Public		CFeeRate& [in] fee double [in] pri
isPriDataPoint() bool Public		CFeeRate& [in] fee double [in] pri
processBlock() void Public		unsigned int [in] nBlockHeight std::vector<CTxMemPoolEntry>& [in] entries bool [in] fCurrentEstimate

Method	Notes	Parameters
processBlockTx() void Public		unsigned int [in] nBlockHeight CTxMemPoolEntry& [in] entry
processTransaction() void Public		CTxMemPoolEntry& [in] entry bool [in] fCurrentEstimate
Read() void Public		CAutoFile& [in] filein
removeTx() void Public		uint256 [in] hash
Write() void Public		CAutoFile& [in] fileout

TxStatsInfo

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {C88512C5-CF43-4c2c-8A7F-C401284DDB13}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TxStatsInfo	Public stats TxConfirmStats	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
blockHeight unsigned int Public		<i>Default:</i>
bucketIndex unsigned int Public		<i>Default:</i>
stats TxConfirmStats* Public		<i>Default:</i>

Operations

Method	Notes	Parameters
TxStatsInfo() Public		

CBlockTemplate**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {F0E6C3BF-BF76-46d1-B7BF-797DF592A451}**Custom Properties**

- isActive = False

Connections

Connector	Source	Target	Notes
Association	Public	Public block	

Connector	Source	Target	Notes
Source -> Destination	CBlockTemplate	CBlock	

Attributes

Attribute	Notes	Constraints and tags
block CBlock Public		<i>Default:</i>
vTxFees std::vector<CAmount> Public		<i>Default:</i>
vTxSigOps std::vector<int64_t> Public		<i>Default:</i>

CBlockTreeDB

Type: Class **CLevelDBWrapper**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {FDC8FEA5-2E0A-40d7-B4BA-39C8C6A7F44D}

Access to the block database (blocks/index/)

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CBlockTreeDB	Public CLevelDBWrapper	

Connector	Source	Target	Notes

Operations

Method	Notes	Parameters
CBlockTreeDB() Public		size_t [in] nCacheSize bool [in] fMemory bool [in] fWipe
CBlockTreeDB() Private		CBlockTreeDB& [in] prm1
LoadBlockIndexGuts() bool Public		
operator=() void Private		CBlockTreeDB& [in] prm1
ReadBlockFileInfo() bool Public		int [in] nFile CBlockFileInfo& [in] fileinfo
ReadFlag() bool Public		std::string& [in] name bool& [in] fValue
ReadLastBlockFile() bool Public		int& [in] nFile
ReadReindexing() bool Public		bool& [in] fReindex
ReadTxIndex() bool Public		uint256& [in] txid CDiskTxPos& [in] pos
WriteBatchSync() bool Public		std::vector<std::pair<int, const CBlockFileInfo*>>& [in] fileInfo int [in] nLastFile std::vector<const CBlockIndex*>& [in] blockinfo
WriteFlag() bool Public		std::string& [in] name bool [in] fValue

Method	Notes	Parameters
WriteReindexing() bool Public		bool [in] fReindex
WriteTxIndex() bool Public		std::vector<std::pair<uint256, CDiskTxPos>>& [in] list

CBlockUndo

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {81481EEA-B6F3-48d0-90CE-85932082BAD9}

Undo information for a CBlock

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_METHODS int Public		<i>Default:</i>
vtxundo std::vector<CTxUndo> Public	for all but the coinbase	<i>Default:</i>

Operations

Method	Notes	Parameters
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action

Method	Notes	Parameters
		int [in] nType int [in] nVersion

CBloomFilter

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model

Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {A98CE94F-502A-4dd0-820D-73CBEE90C1E9}

BloomFilter is a probabilistic filter which SPV clients provide so that we can filter the transactions we send them. This allows for significantly more efficient transaction and block downloads. Because bloom filters are probabilistic, a SPV node can increase the false- positive rate, making us send it transactions which aren't actually its, allowing clients to trade more bandwidth for more privacy by obfuscating which keys are controlled by them.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CNode	Public pfilter CBloomFilter	
Association Source -> Destination	Public CRollingBloomFilter	Private b1 CBloomFilter	
Association Source -> Destination	Public CRollingBloomFilter	Private b2 CBloomFilter	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
isEmpty bool Private		<i>Default:</i>
isFull bool Private		<i>Default:</i>
nFlags unsigned char Private		<i>Default:</i>
nHashFuncs unsigned int Private		<i>Default:</i>
nTweak unsigned int Private		<i>Default:</i>
vData std::vector<unsigned char> Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CBloomFilter()	Private constructor used by	unsigned int [in] nElements

Method	Notes	Parameters
Private	CRollingBloomFilter	double [in] nFPRate unsigned int [in] nTweak
CBloomFilter() Public		unsigned int [in] nElements double [in] nFPRate unsigned int [in] nTweak unsigned char [in] nFlagsIn
CBloomFilter() Public		
clear() void Public		
contains() bool Public		std::vector<unsigned char>& [in] vKey
contains() bool Public		COutPoint& [in] outpoint
contains() bool Public		uint256& [in] hash
Hash() unsigned int Private		unsigned int [in] nHashNum std::vector<unsigned char>& [in] vDataToHash
insert() void Public		std::vector<unsigned char>& [in] vKey
insert() void Public		COutPoint& [in] outpoint
insert() void Public		uint256& [in] hash
IsRelevantAndUpdate() bool Public		CTransaction& [in] tx
IsWithinSizeConstraints() bool Public		
reset() void Public		unsigned int [in] nNewTweak

Method	Notes	Parameters
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
UpdateEmptyFull() void Public		

CBufferedFile

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {82FB643C-EFEE-457d-8970-38D86363B84A}

Non-refcounted RAII wrapper around a FILE* that implements a ring buffer to deserialize from. It guarantees the ability to rewind a given number of bytes. Will automatically close the file when it goes out of scope if not null. If you need to close the file early, use file.fclose() instead of fclose(file).

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBufferedFile	Private nReadLimit uint64_t	
Association Source -> Destination	Public CBufferedFile	Private nReadPos uint64_t	
Association Source -> Destination	Public CBufferedFile	Private nRewind uint64_t	
Association Source -> Destination	Public CBufferedFile	Private nSrcPos uint64_t	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
nReadLimit uint64_t Private	up to which position we're allowed to read	<i>Default:</i>
nReadPos uint64_t Private	how many bytes have been read from this	<i>Default:</i>
nRewind uint64_t Private	how many bytes we guarantee to rewind	<i>Default:</i>
nSrcPos uint64_t Private	how many bytes have been read from source	<i>Default:</i>
nType int Private		<i>Default:</i>
nVersion int Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
src FILE* Private	source file	<i>Default:</i>
vchBuf std::vector<char> Private	the buffer	<i>Default:</i>

Operations

Method	Notes	Parameters
CBufferedFile() Private		CBufferedFile& [in] prm1
CBufferedFile() Public		FILE* [in] fileIn uint64_t [in] nBufSize uint64_t [in] nRewindIn int [in] nTypeIn int [in] nVersionIn
~CBufferedFile() Public		
eof() bool Public		
fclose() void Public		
Fill() bool Protected		
FindByte() void Public		char [in] ch
GetPos() uint64_t Public		
operator=() CBufferedFile& Private		CBufferedFile& [in] prm1
operator>>() CBufferedFile& Public		T& [in] obj
read() CBufferedFile&		char* [in] pch

Method	Notes	Parameters
Public		size_t [in] nSize
Seek() bool Public		uint64_t [in] nPos
SetLimit() bool Public		uint64_t [in] nPos
SetPos() bool Public		uint64_t [in] nPos

CChain

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {E30CCF24-48EC-4cb2-83B8-2C9E3BC4C9D1}

An in-memory indexed chain of blocks.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
vChain std::vector<CBlockIndex*> Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Contains() bool Public		CBlockIndex* [in] pindex
Const FindFork() CBlockIndex * Public		CBlockIndex* [in] pindex
Genesis() CBlockIndex *		

Method	Notes	Parameters
Public		
GetLocator() CBlockLocator Public		CBlockIndex* [in] pindex
Height() int Public		
Next() CBlockIndex * Public		CBlockIndex* [in] pindex
operator[]() CBlockIndex * Public		int [in] nHeight
operator==() bool Public		CChain& [in] a CChain& [in] b
SetTip() void Public	CChain implementation	CBlockIndex* [in] pindex
Tip() CBlockIndex * Public		

CChainParams

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {3C527D79-5C86-4919-BFD6-EADBB88E93C9}

CChainParams defines various tweakable parameters of a given instance of the Bitcoin system. There are three: the main network on which people trade goods and services, the public test network which gets reset from time to time and a regression test mode which is intended for private networks only. It has minimal difficulty to ensure that blocks can be found instantly.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CMainParams	Public CChainParams	
Generalization Source -> Destination	Public CTestNetParams	Public CChainParams	
Generalization Source -> Destination	Public CRegTestParams	Public CChainParams	

Connector	Source	Target	Notes
Association Source -> Destination	Public CChainParams	Protected consensus Params	
Association Source -> Destination	Public CChainParams	Protected nPruneAfterHeight uint64_t	
Association Source -> Destination	Public CChainParams	Protected genesis CBlock	
Association Source -> Destination	Public CChainParams	Protected checkpointData CCheckpointData	

Attributes

Attribute	Notes	Constraints and tags
base58Prefixes std::vector<unsigned char> Protected Collection		<i>Default:</i>
checkpointData CCheckpointData Protected		<i>Default:</i>
consensus Consensus::Params Protected		<i>Default:</i>
fDefaultConsistencyCheck ks bool Protected		<i>Default:</i>

Attribute	Notes	Constraints and tags
fMineBlocksOnDemand bool Protected		<i>Default:</i>
fMiningRequiresPeers bool Protected		<i>Default:</i>
fRequireStandard bool Protected		<i>Default:</i>
fTestnetToBeDeprecated FieldRPC bool Protected		<i>Default:</i>
genesis CBlock Protected		<i>Default:</i>
nDefaultPort int Protected		<i>Default:</i>

Attribute	Notes	Constraints and tags
nPruneAfterHeight uint64_t Protected		<i>Default:</i>
pchMessageStart CMessageHeader::MessageStartChars Protected		<i>Default:</i>
strNetworkID std::string Protected		<i>Default:</i>
vAlertPubKey std::vector<unsigned char> Protected	! Raw pub key bytes for the broadcast alert signing key.	<i>Default:</i>
vFixedSeeds std::vector<SeedSpec6> Protected		<i>Default:</i>
vSeeds std::vector<CDNSSeedData> Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
Const AlertKey() std::vector<unsigned char>& Public		
Const Base58Prefix() std::vector<unsigned char>& Public		Base58Type [in] type
CChainParams() Protected		
Const Checkpoints() CCheckpointData& Public		
DefaultConsistencyCheck() s() bool Public		
Const DNSSeeds() std::vector<CDNSSeedData>& Public		
Const FixedSeeds() std::vector<SeedSpec6>& Public		
Const GenesisBlock() CBlock& Public		
Const GetConsensus() Consensus::Params& Public		
GetDefaultPort() int Public		
Const MessageStart() CMessageHeader::MessageStartChars& Public		
MineBlocksOnDemand() bool Public		
MiningRequiresPeers() bool Public		
NetworkIDString() std::string Public		
PruneAfterHeight() int64_t Public		
RequireStandard() bool Public		
TestnetToBeDeprecatedFieldRPC() bool Public		

Base58Type

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {91499D99-D79D-457b-BB60-0F1B2DDB555B}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
PUBKEY_ADDRESS Public		<i>Default:</i>
SCRIPT_ADDRESS Public		<i>Default:</i>
SECRET_KEY Public		<i>Default:</i>
EXT_PUBLIC_KEY Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
EXT_SECRET_KEY Public		<i>Default:</i>
MAX_BASE58_TYPES Public		<i>Default:</i>

CCheckQueue

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {8D8EBCDF-EA8B-48c5-B5D3-9E10FDCA0AFD}

Queue for verifications that have to be performed. The verifications are represented by a type T, which must provide an operator(), returning a bool. One thread (the master) is assumed to push batches of verifications onto the queue, where they are processed by N-1 worker threads. When the master is done adding work, it temporarily joins the worker pool as an N'th worker, until all jobs are done.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
condMaster boost::condition_variable Private	! Master thread blocks on this when out of work	<i>Default:</i>

Attribute	Notes	Constraints and tags
condWorker boost::condition_variable Private	! Worker threads block on this when out of work	<i>Default:</i>
fAllOk bool Private	! The temporary evaluation result.	<i>Default:</i>
fQuit bool Private	! Whether we're shutting down.	<i>Default:</i>
mutex boost::mutex Private	! Mutex to protect the inner state	<i>Default:</i>
nBatchSize unsigned int Private	! The maximum number of elements to be processed in one batch	<i>Default:</i>
nIdle int Private	! The number of workers (including the master) that are idle.	<i>Default:</i>

Attribute	Notes	Constraints and tags
nTodo unsigned int Private	Number of verifications that haven't completed yet. This includes elements that are no longer queued, but still in the worker's own batches.	<i>Default:</i>
nTotal int Private	! The total number of workers (including the master).	<i>Default:</i>
queue std::vector<T> Private	! The queue of elements to be processed. ! As the order of booleans doesn't matter, it is used as a LIFO (stack)	<i>Default:</i>

Operations

Method	Notes	Parameters
Add() void Public		std::vector<T>& [in] vChecks
CCheckQueue() Public		unsigned int [in] nBatchSizeIn
~CCheckQueue() Public		
IsIdle() bool Public		
Loop() bool Private		bool [in] fMaster
Thread() void Public		
Wait() bool Public		

CCheckQueueControl

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {EF1FA470-44A2-4ab1-BC0B-FAE438C0282D}

RAII-style controller object for a CCheckQueue that guarantees the passed queue is finished before continuing.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
fDone bool Private		<i>Default:</i>
pqueue CCheckQueue<T>* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Add() void Public		<code>std::vector<T>& [in] vChecks</code>
CCheckQueueControl() Public		<code>CCheckQueue<T>* [in] pqueueIn</code>
~CCheckQueueControl() Public		
Wait() bool Public		

CCheckpointData

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {8C8ABF24-06CF-4c78-8950-8353F542A65A}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCheckpointData	Public mapCheckpoints MapCheckpoints	
Association Source -> Destination	Public CCheckpointData	Public nTimeLastCheckpoint int64_t	
Association Source -> Destination	Public CCheckpointData	Public nTransactionsLastChec kpoint int64_t	
Association Source -> Destination	Public CChainParams	Protected checkpointData CCheckpointData	

Attributes

Attribute	Notes	Constraints and tags
fTransactionsPerDay double Public		<i>Default:</i>
mapCheckpoints MapCheckpoints Public		<i>Default:</i>
nTimeLastCheckpoint int64_t Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nTransactionsLastCheck point int64_t Public		<i>Default:</i>

CCClientUIInterface

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {CB5969C2-7CB5-426a-84CF-C4445A5FB131}

Signals for UI communication.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
InitMessage boost::signals2::signal<void (const std::string &message)> Public	Progress message during initialization.	<i>Default:</i>
LoadWallet boost::signals2::signal<void (CWallet* wallet)> Public	A wallet has been loaded.	<i>Default:</i>

Attribute	Notes	Constraints and tags
NotifyAlertChanged boost::signals2::signal<void (const uint256 &hash, ChangeType status)> Public	New, updated or cancelled alert. @note called with lock cs_mapAlerts held.	<i>Default:</i>
NotifyBlockTip boost::signals2::signal<void (const uint256& hash)> Public	New block has been accepted	<i>Default:</i>
NotifyNumConnectionsChanged boost::signals2::signal<void (int newNumConnections)> Public	Number of network connections changed.	<i>Default:</i>
ShowProgress boost::signals2::signal<void (const std::string &title, int nProgress)> Public	Show progress e.g. for verifychain	<i>Default:</i>
ThreadSafeMessageBox boost::signals2::signal<bool (const std::string& message, const std::string& caption, unsigned int style), boost::signals2::last_value<bool> > Public	Show message box.	<i>Default:</i>

MessageBoxFlags

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {ADD692E9-9BFB-43d0-B44E-502192C27440}

Flags for CClientUIInterface::ThreadSafeMessageBox

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
ICON_INFORMATION Public		<i>Default:</i> 0
ICON_WARNING Public		<i>Default:</i> (1U << 0)
ICON_ERROR Public		<i>Default:</i> (1U << 1)
ICON_MASK Public	Mask of all available icons in CClientUIInterface::MessageBoxFlags This needs to be updated, when icons are changed there!	<i>Default:</i> (ICON_INFORMATION ICON_WARNING ICON_ERROR)

Attribute	Notes	Constraints and tags
BTN_OK Public	QMessageBox::Ok	<i>Default:</i> 0x00000400U
BTN_YES Public	QMessageBox::Yes	<i>Default:</i> 0x00004000U
BTN_NO Public	QMessageBox::No	<i>Default:</i> 0x00010000U
BTN_ABORT Public	QMessageBox::Abort	<i>Default:</i> 0x00040000U
BTN_RETRY Public	QMessageBox::Retry	<i>Default:</i> 0x00080000U
BTN_IGNORE Public	QMessageBox::Ignore	<i>Default:</i> 0x00100000U

Attribute	Notes	Constraints and tags
BTN_CLOSE Public	QMessageBox::Close	<i>Default:</i> 0x00200000U
BTN_CANCEL Public	QMessageBox::Cancel	<i>Default:</i> 0x00400000U
BTN_DISCARD Public	QMessageBox::Discard	<i>Default:</i> 0x00800000U
BTN_HELP Public	QMessageBox::Help	<i>Default:</i> 0x01000000U
BTN_APPLY Public	QMessageBox::Apply	<i>Default:</i> 0x02000000U
BTN_RESET Public	QMessageBox::Reset	<i>Default:</i> 0x04000000U

Attribute	Notes	Constraints and tags
BTN_MASK Public	Mask of all available buttons in CClientUIInterface::MessageBoxFlags This needs to be updated, when buttons are changed there!	<i>Default:</i> (BTN_OK BTN_YES BTN_NO BTN_ABORT BTN_RETRY BTN_IGNORE BTN_CLOSE BTN_CANCEL BTN_DISCARD BTN_HELP BTN_APPLY BTN_RESET)
MODAL Public	Force blocking, modal message box dialog (not just OS notification)	<i>Default:</i> 0x10000000U
SECURE Public	Do not print contents of message to debug log	<i>Default:</i> 0x40000000U
MSG_INFORMATION Public	Predefined combinations for certain default usage cases	<i>Default:</i> ICON_INFORMATION
MSG_WARNING Public		<i>Default:</i> (ICON_WARNING BTN_OK MODAL)
MSG_ERROR Public		<i>Default:</i> (ICON_ERROR BTN_OK MODAL)

CCoinControl

Type:	Class
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model
Detail:	<i>Keywords:</i> Created on 8/30/2015. Last modified on 8/30/2015.
GUID:	{18B55373-43FE-4271-9678-49A6F4BE0B56}

Coin Control Features.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCoinControl	Public destChange CTxDestination	

Attributes

Attribute	Notes	Constraints and tags
destChange CTxDestination Public		<i>Default:</i>
fAllowOtherInputs bool Public	! If false, allows unselected inputs, but requires all selected inputs be used	<i>Default:</i>
fAllowWatchOnly bool Public	! Includes watch only addresses which match the ISMINE_WATCH_SOLVABLE criteria	<i>Default:</i>

Attribute	Notes	Constraints and tags
setSelected std::set<COutPoint> Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CCoinControl() Public		
HasSelected() bool Public		
IsSelected() bool Public		uint256& [in] hash unsigned int [in] n
ListSelected() void Public		std::vector<COutPoint>& [in] vOutpoints
Select() void Public		COutPoint& [in] output
SetNull() void Public		
UnSelect() void Public		COutPoint& [in] output
UnSelectAll() void Public		

CCoins**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {5B04DCE0-E407-453f-B67E-32E0D53214B8}

Pruned version of CTransaction: only retains metadata and unspent transaction outputs Serialized format: - VARINT(nVersion) - VARINT(nCode) - unspentness bitvector, for vout[2] and further; least significant byte first - the non-spent CTxOuts (via CTxOutCompressor) - VARINT(nHeight) The nCode value consists of: - bit 1: IsCoinBase() - bit 2: vout[0] is not spent - bit 4: vout[1] is not spent - The higher bits encode N, the number of non-zero bytes in the following bitvector. - In case both bit 2 and bit 4 are unset, they encode N-1, as there must be at least one non-spent output). Example: 0104835800816115944e077fe7c803cfa57f29b36bf87c1d358bb85e

<><><-----><----> | \ | | /
version code vout[1] height - version = 1 - code = 4 (vout[1] is not spent, and 0 non-zero bytes

of bitvector follow) - unspentness bitvector: as 0 non-zero bytes follow, it has length 0 - vout[1]:
835800816115944e077fe7c803cfa57f29b36bf87c1d35 8358: compact amount representation for 60000000000 (600 BTC)
00: special txout type pay-to-pubkey-hash 816115944e077fe7c803cfa57f29b36bf87c1d35: address uint160 - height = 203998 Example:
0109044086ef97d5790061b01caab50f1b8e9c50a5057eb43c2d9563a4eebbd123008c988f1a4a4de2161e0f50aac7f17e7f9555caa486af3b
<><><-----><-----><---->
/ \ | | /
version code unspentness vout[4] vout[16] height - version = 1 - code
= 9 (coinbase, neither vout[0] or vout[1] are unspent, 2 (1, +1 because both bit 2 and bit 4 are unset) non-zero bitvector bytes follow) - unspentness bitvector: bits 2 (0x04) and 14 (0x4000) are set, so vout[2+2] and vout[14+2] are unspent - vout[4]: 86ef97d5790061b01caab50f1b8e9c50a5057eb43c2d9563a4ee 86ef97d579: compact amount representation for 234925952 (2.35 BTC)
00: special txout type pay-to-pubkey-hash 61b01caab50f1b8e9c50a5057eb43c2d9563a4ee: address uint160 - vout[16]: bbd123008c988f1a4a4de2161e0f50aac7f17e7f9555caa4 bbd123: compact amount representation for 110397 (0.001 BTC)
00: special txout type pay-to-pubkey-hash 8c988f1a4a4de2161e0f50aac7f17e7f9555caa4: address uint160 - height = 120891

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCoinsCacheEntry	Public coins CCoins	

Attributes

Attribute	Notes	Constraints and tags
fCoinBase bool Public	! whether transaction is a coinbase	<i>Default:</i>
nHeight int Public	! at which height this transaction was included in the active block chain	<i>Default:</i>

Attribute	Notes	Constraints and tags
nVersion int Public	! version of the CTransaction; accesses to this value should probably check for nHeight as well, ! as new tx version will probably only be introduced at certain heights	<i>Default:</i>
vout std::vector<CTxOut> Public	! unspent transaction outputs; spent outputs are .IsNull(); spent outputs at the end of the array are dropped	<i>Default:</i>

Operations

Method	Notes	Parameters
CalcMaskSize() void Public	calculate number of bytes for the bitmask, and its number of non-zero bytes each bit in the bitmask represents the availability of one output, but the availabilities of the first two outputs are encoded separately	unsigned int& [in] nBytes unsigned int& [in] nNonzeroBytes
CCoins() Public		CTransaction& [in] tx int [in] nHeightIn
CCoins() Public		
Cleanup() void Public		
Clear() void Public		
ClearUnspendable() void Public		
DynamicMemoryUsage() size_t Public		
FromTx() void Public		CTransaction& [in] tx int [in] nHeightIn
GetSerializeSize() unsigned int Public		int [in] nType int [in] nVersion
IsAvailable() bool Public		unsigned int [in] nPos

Method	Notes	Parameters
IsCoinBase() bool Public		
IsPruned() bool Public		
operator!=() bool Public		CCoins& [in] a CCoins& [in] b
operator==() bool Public		CCoins& [in] a CCoins& [in] b
Serialize() void Public		Stream& [in] s int [in] nType int [in] nVersion
Spend() bool Public		uint32_t [in] nPos
swap() void Public		CCoins& [in] to
Unserialize() void Public		Stream& [in] s int [in] nType int [in] nVersion

CCoinsCacheEntry

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {25291C5B-54C7-4a2e-A674-29BE9CFEA4DB}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCoinsCacheEntry	Public coins CCoins	

Attributes

Attribute	Notes	Constraints and tags
coins CCoin Public	The actual cached data.	<i>Default:</i>
flags unsigned char Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CCoinsCacheEntry() Public		

Flags

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {3238CB05-924C-47a0-A570-661C8FA34730}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
DIRTY Public	This cache entry is potentially different from the version in the parent view.	<i>Default:</i> (1 << 0)
FRESH Public	The parent view does not have this entry (or it is pruned).	<i>Default:</i> (1 << 1)

CCoinsKeyHasher

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {E012AD88-DDBB-42f3-A810-966649E679FE}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCoinsKeyHasher	Private salt uint256	

Attributes

Attribute	Notes	Constraints and tags
salt uint256 Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CCoinsKeyHasher() Public		
operator()() size_t Public		uint256& [in] key

CCoinsMap

Type: **Class** **boost::unordered_map<uint256, CCoinCacheEntry, CCoinKeyHasher>**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {F024CEA1-127C-4006-B680-7A0940545F67}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCoinsViewCache	Protected cacheCoins CCoinsMap	

CCoinsModifier

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {244DADB0-BFB6-4877-AF24-F06C9DEDD8B5}

A reference to a mutable cache entry. Encapsulating it allows us to run cleanup code after the modification is finished, and keeping track of concurrent modifications.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
-----------	--------	--------	-------

Connector	Source	Target	Notes
Association Source -> Destination	Public CCoinsModifier	Private cache CCoinsViewCache	

Attributes

Attribute	Notes	Constraints and tags
cache CCoinsViewCache& Private		<i>Default:</i>
cachedCoinUsage size_t Private	Cached memory usage of the CCoin object before modification	<i>Default:</i>
it CCoinsMap::iterator Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CCoinsModifier() Private		CCoinsViewCache& [in] cache_ CCoinsMap::iterator [in] it_ size_t [in] usage
~CCoinsModifier() Public		
operator->() CCoin* Public		
operator*() CCoin& Public		

CCoinsStatsType: Class

Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {3F0B2307-9694-4d44-9B9A-F8BD5BB106C4}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCoinsStats	Public nTotalAmount CAmount	
Association Source -> Destination	Public CCoinsStats	Public hashBlock uint256	
Association Source -> Destination	Public CCoinsStats	Public nSerializedSize uint64_t	
Association Source -> Destination	Public CCoinsStats	Public nTransactionOutputs uint64_t	
Association Source -> Destination	Public CCoinsStats	Public nTransactions uint64_t	
Association Source -> Destination	Public CCoinsStats	Public hashSerialized uint256	

Attributes

Attribute	Notes	Constraints and tags
hashBlock uint256 Public		<i>Default:</i>
hashSerialized uint256 Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nHeight int Public		<i>Default:</i>
nSerializedSize uint64_t Public		<i>Default:</i>
nTotalAmount CAmount Public		<i>Default:</i>
nTransactionOutputs uint64_t Public		<i>Default:</i>
nTransactions uint64_t Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CCoinsStats() Public		

CCoinsView*Type:*Class*Status:*

Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {F6152171-DB14-45a4-A897-46DEB9D79C22}

Abstract view on the open txout dataset.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CCoinsViewDB	Public CCoinsView	
Association Source -> Destination	Public CCoinsViewBacked	Protected base CCoinsView	
Generalization Source -> Destination	Public CCoinsViewBacked	Public CCoinsView	

Operations

Method	Notes	Parameters
BatchWrite() bool Public		CCoinsMap& [in] mapCoins uint256& [in] hashBlock
~CCoinsView() Public		
GetBestBlock() uint256 Public		
GetCoins() bool Public		uint256& [in] txid CCoins& [in] coins
GetStats() bool Public		CCoinsStats& [in] stats
HaveCoins() bool Public		uint256& [in] txid

CCoinsViewBacked

Type: **Class** CCoinsView
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {E95FF9FB-7206-42ab-9C6E-0AF6AA71688B}

CCoinsView backed by another CCoinsView

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CCoinsViewMemPool	Public CCoinsViewBacked	
Association Source -> Destination	Public CCoinsViewBacked	Protected base CCoinsView	
Generalization Source -> Destination	Public CCoinsViewBacked	Public CCoinsView	
Generalization Source -> Destination	Public CCoinsViewCache	Public CCoinsViewBacked	

Attributes

Attribute	Notes	Constraints and tags
base CCoinsView* Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
BatchWrite() bool Public		CCoinsMap& [in] mapCoins uint256& [in] hashBlock
CCoinsViewBacked() Public		CCoinsView* [in] viewIn
GetBestBlock() uint256 Public		
GetCoins() bool Public		uint256& [in] txid CCoins& [in] coins

Method	Notes	Parameters
GetStats() bool Public		CCoinsStats& [in] stats
HaveCoins() bool Public		uint256& [in] txid
SetBackend() void Public		CCoinsView& [in] viewIn

CCoinsViewCache

Type: **Class** [CCoinsViewBacked](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model

Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {A7A83F56-713A-4c7b-BD80-4EC029E3C8CA}

CCoinsView that adds a memory cache for transactions to another CCoinView

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCoinsModifier	Private cache CCoinsViewCache	
Association Source -> Destination	Public CCoinsViewCache	Protected cacheCoins CCoinsMap	
Association Source -> Destination	Public CCoinsViewCache	Protected hashBlock uint256	
Generalization Source -> Destination	Public CCoinsViewCache	Public CCoinsViewBacked	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
cacheCoins CCoinsMap Protected		<i>Default:</i> [<u>mutable</u> = true]
cachedCoinsUsage size_t Protected	Cached dynamic memory usage for the inner CCoins objects.	<i>Default:</i> [<u>mutable</u> = true]
hashBlock uint256 Protected	Make mutable so that we can "fill the cache" even from Get-methods declared as "const".	<i>Default:</i> [<u>mutable</u> = true]
hasModifier bool Protected	Whether this cache has an active modifier.	<i>Default:</i>

Operations

Method	Notes	Parameters
Const AccessCoins() CCoins* Public		uint256& [in] txid
BatchWrite() bool Public		CCoinsMap& [in] mapCoins uint256& [in] hashBlock
CCoinsViewCache() Public		CCoinsView* [in] baseIn
~CCoinsViewCache() Public		
CCoinsViewCache() Private		CCoinsViewCache& [in] prm1
DynamicMemoryUsage()		

Method	Notes	Parameters
size_t Public		
FetchCoins() CCoinsMap::iterator Private		uint256& [in] txid
FetchCoins() CCoinsMap::const_iterator Private		uint256& [in] txid
Flush() bool Public		
GetBestBlock() uint256 Public		
GetCacheSize() unsigned int Public		
GetCoins() bool Public		uint256& [in] txid CCoins& [in] coins
Const GetOutputFor() CTxOut & Public		CTxIn& [in] input
GetPriority() double Public		CTransaction& [in] tx int [in] nHeight
GetValueIn() CAmount Public		CTransaction& [in] tx
HaveCoins() bool Public		uint256& [in] txid
HaveInputs() bool Public		CTransaction& [in] tx
ModifyCoins() CCoinsModifier Public		uint256& [in] txid
SetBestBlock() void Public		uint256& [in] hashBlock

CCoinsViewDB

Type: Class **CCoinsView**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {AA753CC4-A2BB-42e1-A92B-FF8B3054D059}

CCoinsView backed by the LevelDB coin database (chainstate/)

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TestingSetup	Public pcoinsdbview CCoinsViewDB	
Association Source -> Destination	Public CCoinsViewDB	Protected db CLevelDBWrapper	
Generalization Source -> Destination	Public CCoinsViewDB	Public CCoinsView	

Attributes

Attribute	Notes	Constraints and tags
db CLevelDBWrapper Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
BatchWrite() bool Public		CCoinsMap& [in] mapCoins uint256& [in] hashBlock
CCoinsViewDB() Public		size_t [in] nCacheSize bool [in] fMemory bool [in] fWipe
GetBestBlock() uint256 Public		
GetCoins() bool Public		uint256& [in] txid CCoins& [in] coins
GetStats() bool		CCoinsStats& [in] stats

Method	Notes	Parameters
Public		
HaveCoins() bool Public		uint256& [in] txid

CCoinsViewMemPool

Type: Class **CCoinsViewBacked**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {175FA327-5C99-41b3-A425-AD38ADC5A0D6}

CCoinsView that brings transactions from a mempool into view. It does not check for spendings by memory pool transactions.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCoinsViewMemPool	Protected mempool CTxMemPool	
Generalization Source -> Destination	Public CCoinsViewMemPool	Public CCoinsViewBacked	

Attributes

Attribute	Notes	Constraints and tags
mempool CTxMemPool& Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
CCoinsViewMemPool() Public		CCoinsView* [in] baseIn CTxMemPool& [in] mempoolIn

Method	Notes	Parameters
GetCoins() bool Public		uint256& [in] txid CCoins& [in] coins
HaveCoins() bool Public		uint256& [in] txid

CCrypter

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {E1EEF17D-3020-4b76-91A6-B57330083E28}

Encryption/decryption context with key information

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
chIV unsigned char Private Collection		<i>Default:</i>
chKey unsigned char Private Collection		<i>Default:</i>
fKeySet bool Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CCrypter() Public		
~CCrypter() Public		
CleanKey() void Public		
Decrypt() bool Public		std::vector<unsigned char>& [in] vchCiphertext CKeyingMaterial& [in] vchPlaintext
Encrypt() bool Public		CKeyingMaterial& [in] vchPlaintext std::vector<unsigned char>& [in] vchCiphertext
SetKey() bool Public		CKeyingMaterial& [in] chNewKey std::vector<unsigned char>& [in] chNewIV
SetKeyFromPassphrase() bool Public		SecureString& [in] strKeyData std::vector<unsigned char>& [in] chSalt unsigned int [in] nRounds unsigned int [in] nDerivationMethod

CCryptoKeyStore

Type: **Class** **CBasicKeyStore**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {2213D1B6-F966-4947-9C08-363EEE6E0C2F}

Keystore which keeps the private keys encrypted. It derives from the basic key store, which is used if no encryption is active.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCryptoKeyStore	Private mapCryptedKeys CryptedKeyMap	
Association Source -> Destination	Public CCryptoKeyStore	Private vMasterKey CKeyingMaterial	
Generalization Source -> Destination	Public CWallet	Public CCryptoKeyStore	
Generalization Source -> Destination	Public CCryptoKeyStore	Public CBasicKeyStore	

Attributes

Attribute	Notes	Constraints and tags
fDecryptionThoroughlyChecked bool Private	! keeps track of whether Unlock has run a thorough check before	<i>Default:</i>
fUseCrypto bool Private	! if fUseCrypto is true, mapKeys must be empty ! if fUseCrypto is false, vMasterKey must be empty	<i>Default:</i>
mapCryptedKeys CryptedKeyMap Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
NotifyStatusChanged boost::signals2::signal<void (CCryptoKeyStore* wallet)> Public	Wallet status (encrypted, locked) changed. Note: Called without locks held.	<i>Default:</i>
vMasterKey CKeyingMaterial Private		<i>Default:</i>

Operations

Method	Notes	Parameters
AddCryptedKey() bool Public		CPubKey& [in] vchPubKey std::vector<unsigned char>& [in] vchCryptedSecret
AddKeyPubKey() bool Public		CKey& [in] key CPubKey& [in] pubkey
CCryptoKeyStore() Public		
EncryptKeys() bool Protected		CKeyingMaterial& [in] vMasterKeyIn
GetKey() bool Public		CKeyID& [in] address CKey& [in] keyOut
GetKeys() void Public		std::set<CKeyID>& [in] setAddress
GetPubKey() bool Public		CKeyID& [in] address CPubKey& [in] vchPubKeyOut
HaveKey() bool		CKeyID& [in] address

Method	Notes	Parameters
Public		
IsCrypted() bool Public		
IsLocked() bool Public		
Lock() bool Public		
SetCrypted() bool Protected		
Unlock() bool Protected		CKeyingMaterial& [in] vMasterKeyIn

CDB

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {665FCD75-0639-4ba5-8862-C12A126D2F57}

RAII class that provides access to a Berkeley database

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CWalletDB	Public CDB	

Attributes

Attribute	Notes	Constraints and tags
activeTxn DbTxn* Protected		<i>Default:</i>

Attribute	Notes	Constraints and tags
fFlushOnClose bool Protected		<i>Default:</i>
fReadOnly bool Protected		<i>Default:</i>
pdb Db*		<i>Default:</i>
Protected		
strFile std::string Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
CDB() Protected		std::string& [in] strFilename char* [in] pszMode bool [in] fFlushOnCloseIn
~CDB() Protected		
CDB() Private		CDB& [in] prm1
Close() void Public		
Erase() bool Protected		K& [in] key

Method	Notes	Parameters
Exists() bool Protected		K& [in] key
Flush() void Public		
GetCursor() Dbc* Protected		
operator=() void Private		CDB& [in] prm1
Read() bool Protected		K& [in] key T& [in] value
ReadAtCursor() int Protected		Dbc* [in] pcursor CDataStream& [in] ssKey CDataStream& [in] ssValue unsigned int [in] fFlags
ReadVersion() bool Public		int& [in] nVersion
Static Rewrite() bool Public		std::string& [in] strFile char* [in] pszSkip
TxnAbort() bool Public		
TxnBegin() bool Public		
TxnCommit() bool Public		
Write() bool Protected		K& [in] key T& [in] value bool [in] fOverwrite
WriteVersion() bool Public		int [in] nVersion

CDBEnv

Type:

Class

Status:

Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {EE2302F4-503F-4864-A093-B5BA1740A694}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
cs_db CCriticalSection Public		<i>Default:</i> [mutable = true]
dbenv DbEnv*		<i>Default:</i>
fDbEnvInit bool Private		<i>Default:</i>
fMockDb bool Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
mapDb std::map<std::string, Db*> Public		<i>Default:</i>
mapFileUseCount std::map<std::string, int> Public		<i>Default:</i>
strPath std::string Private	Don't change into boost::filesystem::path, as that can result in shutdown problems/crashes caused by a static initialized internal pointer.	<i>Default:</i>

Operations

Method	Notes	Parameters
CDBEnv() Public		
~CDBEnv() Public		
CheckpointLSN() void Public		std::string& [in] strFile
Close() void Public		
CloseDb() void Public		std::string& [in] strFile
EnvShutdown() void Private		
Flush() void Public		bool [in] fShutdown
IsMock() bool Public		
MakeMock() void Public		
Open() bool Public		boost::filesystem::path& [in] path
RemoveDb() bool		std::string& [in] strFile

Method	Notes	Parameters
Public		
Reset() void Public		
Salvage() bool Public		std::string& [in] strFile bool [in] fAggressive std::vector<KeyValPair>& [in] vResult
TxnBegin() DbTxn* Public		int [in] flags
Verify() VerifyResult Public		std::string& [in] strFile bool [in] prm1

KeyValPair

Type: Class **std::pair<std::vector<unsigned char>, std::vector<unsigned char> >**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {976D9F33-9FFE-4ff9-AEE6-82C0C6651720}

Salvage data from a file that Verify says is bad. fAggressive sets the DB_AGGRESSIVE flag (see berkeley DB->verify() method documentation). Appends binary key/value pairs to vResult, returns true if successful. NOTE: reads the entire database into memory, so cannot be used for huge databases.

Custom Properties

- isActive = False

VerifyResult

Type: Enumeration
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {51ED23F6-A072-4be6-84A8-719C5A3E5D33}

Verify that database file strFile is OK. If it is not, call the callback to try to recover. This must be called BEFORE strFile is opened. Returns true if strFile is OK.

Custom Properties

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
VERIFY_OK Public		<i>Default:</i>
RECOVER_OK Public		<i>Default:</i>
RECOVER_FAIL Public		<i>Default:</i>

CDNSSeedData

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {73C2B5FE-6B1E-40e5-92B3-4B06C8D12984}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
host std::string Public		<i>Default:</i>
name std::string Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CDNSSeedData() Public		std::string& [in] strName std::string& [in] strHost

CDataStream**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {EAE24A86-B6B8-47b5-B0FC-AF15D7527D4E}

Double ended buffer combining vector and stream-like interfaces. >> and << read and write unformatted data using the above serialization templates. Fills with data in linear time; some stringstream implementations take N^2 time.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CDataStream	Protected vch vector_type	
Association Source -> Destination	Public CNetMessage	Public hdrbuf CDataStream	

Connector	Source	Target	Notes
Association Source -> Destination	Public CNetMessage	Public vRecv CDataStream	
Association Source -> Destination	Public CNode	Public ssSend CDataStream	

Attributes

Attribute	Notes	Constraints and tags
nReadPos unsigned int Protected		<i>Default:</i>
nType int Public		<i>Default:</i>
nVersion int Public		<i>Default:</i>
vch vector_type Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
begin() const_iterator Public		
begin() iterator Public		
CDataStream() Public		int [in] nTypeIn int [in] nVersionIn

Method	Notes	Parameters
CDataStream() Public		const_iterator [in] pbegin const_iterator [in] pend int [in] nTypeIn int [in] nVersionIn
CDataStream() Public		char* [in] pbegin char* [in] pend int [in] nTypeIn int [in] nVersionIn
CDataStream() Public		vector_type& [in] vchIn int [in] nTypeIn int [in] nVersionIn
CDataStream() Public		std::vector<char>& [in] vchIn int [in] nTypeIn int [in] nVersionIn
CDataStream() Public		std::vector<unsigned char>& [in] vchIn int [in] nTypeIn int [in] nVersionIn
clear() void Public		
Compact() void Public		
empty() bool Public		
end() const_iterator Public		
end() iterator Public		
eof() bool Public		
erase() iterator		iterator [in] it

Method	Notes	Parameters
Public		
erase() iterator Public		iterator [in] first iterator [in] last
GetAndClear() void Public		CSerializeData& [in] data
GetSerializeSize() unsigned int Public		T& [in] obj
GetType() int Public		
GetVersion() int Public		
ignore() CDataStream& Public		int [in] nSize
in_avail() int Public		
Init() void Public		int [in] nTypeIn int [in] nVersionIn
insert() iterator Public		iterator [in] it char& [in] x
insert() void Public		iterator [in] it size_type [in] n char& [in] x
insert() void Public		iterator [in] it std::vector<char>::const_iterator [in] first std::vector<char>::const_iterator [in] last
insert() void Public		iterator [in] it char* [in] first char* [in] last

Method	Notes	Parameters
operator[]() const_reference Public		size_type [in] pos
operator[]() reference Public		size_type [in] pos
operator+() CDataStream Public		CDataStream& [in] a CDataStream& [in] b
operator+=() CDataStream& Public		CDataStream& [in] b
operator<<() CDataStream& Public		T& [in] obj
operator>>() CDataStream& Public		T& [in] obj
rdbuf() CDataStream*		
Public		
read() CDataStream& Public		char* [in] pch size_t [in] nSize
ReadVersion() void Public		
reserve() void Public		size_type [in] n
resize() void Public		size_type [in] n value_type [in] c
Rewind() bool Public		size_type [in] n
Serialize() void Public		Stream& [in] s int [in] nType int [in] nVersion
SetType() void Public		int [in] n
SetVersion() void Public		int [in] n
size() size_type		

Method	Notes	Parameters
Public		
str() std::string		
Public		
write() CDataStream&		char* [in] pch
Public		size_t [in] nSize
WriteVersion() void		
Public		

allocator_type

Type: **Class** [vector_type::allocator_type](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {BD71FD48-21CB-44f4-B117-6DE52B2E0B04}

Custom Properties

- isActive = False

const_iterator

Type: **Class** [vector_type::const_iterator](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D9D7EB85-9D63-403a-B132-5730A0087145}

Custom Properties

- isActive = False

const_reference

Type: **Class** [vector_type::const_reference](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {DC3532E8-F1B7-41f9-A3EA-9A635A492B7F}

Custom Properties

- isActive = False

difference_type

Type: **Class** **vector_type::difference_type**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {CE1C34EC-C667-443c-B5FC-111E908FEC41}

Custom Properties

- isActive = False

iterator

Type: **Class** **vector_type::iterator**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {FBEE53BA-637A-4377-99C2-F03B9F0162BC}

Custom Properties

- isActive = False

reference

Type: **Class** **vector_type::reference**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {320B1139-3E96-477c-8074-E4E919FD2B56}

Custom Properties

- isActive = False

reverse_iterator

Type: **Class** [vector_type::reverse_iterator](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {F9203CD3-3071-46a3-81F4-A9C60A0FF8AC}

Custom Properties

- isActive = False

size_type

Type: **Class** [vector_type::size_type](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {177BCDBD-56F2-4e0b-83F2-C11D91B4814D}

Custom Properties

- isActive = False

value_type

Type: **Class** [vector_type::value_type](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {5C8FC6EE-3A1C-4ac1-988A-6B23CFC5DD99}

Custom Properties

- isActive = False

vector_type

Type: [Class](#) [CSerializeData](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {5BC7686C-155F-49b5-B430-7B1CD94B2EA4}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CDataStream	Protected vch vector_type	
Generalization Source -> Destination	Public vector_type	Public CSerializeData	

CDiskBlockIndex

Type: [Class](#) [CBlockIndex](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {BB83A542-55E8-4813-89C9-C1B7914188AA}

Used to marshal pointers into hashes for db storage.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CDiskBlockIndex	Public hashPrev uint256	
Generalization Source -> Destination	Public CDiskBlockIndex	Public CBlockIndex	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
hashPrev uint256 Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CDiskBlockIndex() Public		
CDiskBlockIndex() Public		CBlockIndex* [in] pindex
GetBlockHash() uint256 Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
ToString() std::string Public		

CDiskBlockPos

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {3C056C12-D37E-42a4-ADBD-6EB9FFD2C94B}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CDiskTxPos	Public CDiskBlockPos	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
nFile int Public		<i>Default:</i>
nPos unsigned int Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CDiskBlockPos() Public		
CDiskBlockPos() Public		int [in] nFileIn unsigned int [in] nPosIn
IsNull() bool Public		
operator!=() bool Public		CDiskBlockPos& [in] a CDiskBlockPos& [in] b

Method	Notes	Parameters
operator==() bool Public		CDiskBlockPos& [in] a CDiskBlockPos& [in] b
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		
ToString() std::string Public		

CDiskTxPos

Type: Class **CDiskBlockPos**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {64CEE262-03BF-4d97-8191-E0982F58C0A1}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CDiskTxPos	Public CDiskBlockPos	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
nTxOffset unsigned int Public	after header	<i>Default:</i>

Operations

Method	Notes	Parameters
CDiskTxPos() Public		CDiskBlockPos& [in] blockIn unsigned int [in] nTxOffsetIn
CDiskTxPos() Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		

CECKey**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {6A09C248-FA6B-44f3-AAC3-A912780CCC8F}

RAII Wrapper around OpenSSL's EC_KEY

Custom Properties

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
pkey EC_KEY*	Private	<i>Default:</i>

Operations

Method	Notes	Parameters
CECKey() Public	anon namespace	
~CECKey() Public		
GetPubKey() void Public		std::vector<unsigned char>& [in] pubkey bool [in] fCompressed
Recover() bool Public		uint256& [in] hash unsigned char* [in] p64 int [in] rec
Static SanityCheck() bool Public		
SetPubKey() bool Public		unsigned char* [in] pubkey size_t [in] size
TweakPublic() bool Public		unsigned char [in] vchTweak[32]
Verify() bool Public		uint256& [in] hash std::vector<unsigned char>& [in] vchSig

CExtKey

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {0D23DC22-AE93-426f-B7DB-AD03D46EB0AF}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CExtKey	Public chaincode ChainCode	
Association Source -> Destination	Public CExtKey	Public key CKey	

Attributes

Attribute	Notes	Constraints and tags
chaincode ChainCode Public		<i>Default:</i>
key CKey Public		<i>Default:</i>
nChild unsigned int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nDepth unsigned char Public		<i>Default:</i>
vchFingerprint unsigned char Public Collection		<i>Default:</i>

Operations

Method	Notes	Parameters
Decode() void Public		unsigned char [in] code[74]
Derive() bool Public		CExtKey& [in] out unsigned int [in] nChild
Encode() void Public		unsigned char [in] code[74]
Neuter() CExtPubKey Public		
operator==() bool Public		CExtKey& [in] a CExtKey& [in] b
SetMaster() void Public		unsigned char* [in] seed unsigned int [in] nSeedLen

CExtPubKey**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {EC3A0A08-A3E5-401d-B88B-36A601699A95}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CExtPubKey	Public chaincode ChainCode	
Association Source -> Destination	Public CExtPubKey	Public pubkey CPubKey	

Attributes

Attribute	Notes	Constraints and tags
chaincode ChainCode Public		<i>Default:</i>
nChild unsigned int Public		<i>Default:</i>
nDepth unsigned char Public		<i>Default:</i>
pubkey CPubKey Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
vchFingerprint unsigned char Public Collection		<i>Default:</i>

Operations

Method	Notes	Parameters
Decode() void Public		unsigned char [in] code[74]
Derive() bool Public		CExtPubKey& [in] out unsigned int [in] nChild
Encode() void Public		unsigned char [in] code[74]
operator==() bool Public		CExtPubKey& [in] a CExtPubKey& [in] b

CFeeRate**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {8A3AE900-E8F3-4f2d-9C4C-43C8C4FF01AC}

Type-safe wrapper class to for fee rates (how much to pay based on transaction size)

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBlockPolicyEstimator	Private feeUnlikely CFeeRate	
Association Source -> Destination	Public CBlockPolicyEstimator	Private minTrackedFee CFeeRate	

Connector	Source	Target	Notes
Association Source -> Destination	Public CBlockPolicyEstimator	Private feeLikely CFeeRate	
Association Source -> Destination	Public COrphan	Public feeRate CFeeRate	
Association Source -> Destination	Public CFeeRate	Private nSatoshisPerK CAmount	
Association Source -> Destination	Public CWallet	Public minTxFee CFeeRate	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_METHODS int Public		<i>Default:</i>
nSatoshisPerK CAmount Private	unit is satoshis-per-1,000-bytes	<i>Default:</i>

Operations

Method	Notes	Parameters
CFeeRate() Public		
CFeeRate() Public		CAmount& [in] _nSatoshisPerK
CFeeRate() Public		CAmount& [in] nFeePaid size_t [in] nSize
CFeeRate() Public		CFeeRate& [in] other
GetFee() CAmount Public		size_t [in] size
GetFeePerK() CAmount		

Method	Notes	Parameters
Public operator<() bool		CFeeRate& [in] a CFeeRate& [in] b
Public operator<=() bool		CFeeRate& [in] a CFeeRate& [in] b
Public operator==() bool		CFeeRate& [in] a CFeeRate& [in] b
Public operator>() bool		CFeeRate& [in] a CFeeRate& [in] b
Public operator>=() bool		CFeeRate& [in] a CFeeRate& [in] b
Public SerializationOp() void		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
Public ToString() std::string		

CFlatData

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {422EE469-8BFD-42c5-BE2D-F63284D5614A}

Wrapper for serializing arrays and POD.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
pbegin char*		<i>Default:</i>
Protected		

Attribute	Notes	Constraints and tags
pend char*		<i>Default:</i>
Protected		

Operations

Method	Notes	Parameters
begin() char*		
Public		
Const begin() char*		
Public		
CFlatData()		void* [in] pbeginIn
Public		void* [in] pendIn
CFlatData()		std::vector<T,TAl>& [in] v
end() char*		
Public		
Const end() char*		
Public		
GetSerializeSize()		int [in] prm1
unsigned int		int [in] prm2
Public		
Serialize() void		Stream& [in] s
Public		int [in] prm1
		int [in] prm2
Unserialize() void		Stream& [in] s
Public		int [in] prm1

Method	Notes	Parameters
		int [in] prm2

CHMAC_SHA256

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {60F62D73-A369-4e2f-8A3C-AFC315B4CF17}

A hasher class for HMAC-SHA-512.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CHMAC_SHA256	Private inner CSHA256	
Association Source -> Destination	Public CHMAC_SHA256	Private outer CSHA256	

Attributes

Attribute	Notes	Constraints and tags
inner CSHA256 Private		<i>Default:</i>
outer CSHA256 Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
OUTPUT_SIZE size_t Public Static Const		<i>Default:</i> 32

Operations

Method	Notes	Parameters
CHMAC_SHA256() Public		unsigned char* [in] key size_t [in] keylen
Finalize() void Public		unsigned char [in] hash[OUTPUT_SIZE]
Write() CHMAC_SHA256& Public		unsigned char* [in] data size_t [in] len

CHMAC_SHA512

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {016AEAF0-B42C-452a-9F02-BBAEB2814A3F}

A hasher class for HMAC-SHA-512.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CHMAC_SHA512	Private inner CSHA512	
Association Source -> Destination	Public CHMAC_SHA512	Private outer CSHA512	

Attributes

Attribute	Notes	Constraints and tags
inner CSHA512 Private		<i>Default:</i>
outer CSHA512 Private		<i>Default:</i>
OUTPUT_SIZE size_t Public Static Const		<i>Default:</i> 64

Operations

Method	Notes	Parameters
CHMAC_SHA512() Public		unsigned char* [in] key size_t [in] keylen
Finalize() void Public		unsigned char [in] hash[OUTPUT_SIZE]
Write() CHMAC_SHA512& Public		unsigned char* [in] data size_t [in] len

CHash160

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {9F07BB1E-CC03-4e35-87FF-82ECCEF7C723}

A hasher class for Bitcoin's 160-bit hash (SHA-256 + RIPEMD-160).

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CHash160	Private sha CSHA256	

Attributes

Attribute	Notes	Constraints and tags
OUTPUT_SIZE size_t Public Static Const		<i>Default:</i> CRIPEMD160::OUTPUT_SIZE
sha CSHA256 Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Finalize() void Public		unsigned char [in] hash[OUTPUT_SIZE]
Reset() CHash160& Public		
Write() CHash160& Public		unsigned char* [in] data size_t [in] len

CHash256

Type:	Class
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>
GUID:	{749ED4F3-880E-450b-883C-1D434733FFD7}

A hasher class for Bitcoin's 256-bit hash (double SHA-256).

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CHashWriter	Private ctx CHash256	
Association Source -> Destination	Public CHash256	Private sha CSHA256	

Attributes

Attribute	Notes	Constraints and tags
OUTPUT_SIZE size_t Public Static Const		<i>Default:</i> CSHA256::OUTPUT_SIZE
sha CSHA256 Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Finalize() void Public		unsigned char [in] hash[OUTPUT_SIZE]
Reset() CHash256& Public		
Write() CHash256& Public		unsigned char* [in] data

Method	Notes	Parameters
		size_t [in] len

CHashWriter

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {63EDEABE-6849-46b2-ACF9-234F7DC2269B}

A writer stream (for serialization) that computes a 256-bit hash.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CHashWriter	Private ctx CHash256	

Attributes

Attribute	Notes	Constraints and tags
ctx CHash256 Private		<i>Default:</i>
nType int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nVersion int Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CHashWriter() Public		int [in] nTypeIn int [in] nVersionIn
GetHash() uint256 Public		
operator<<() CHashWriter& Public		T& [in] obj
write() CHashWriter& Public		char* [in] pch size_t [in] size

CInPoint**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {6D3CE63D-61DF-41f7-BD85-8A3FDB6A2008}

An inpoint - a combination of a transaction and an index n into its vin

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CInPoint	Public n uint32_t	
Association Source -> Destination	Public CInPoint	Public ptx CTransaction	

Attributes

Attribute	Notes	Constraints and tags
n uint32_t Public		<i>Default:</i>
ptx CTransaction* Public Const		<i>Default:</i>

Operations

Method	Notes	Parameters
CInPoint() Public		
CInPoint() Public		CTransaction* [in] ptxIn uint32_t [in] nIn
DynamicMemoryUsage() size_t Public		
IsNull() bool Public		
SetNull() void Public		

Clv

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {5D836799-1DB7-4098-953A-EB9ADAED9683}

inv message data

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CInv	Public hash uint256	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
hash uint256 Public		<i>Default:</i>
type int Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CInv() Public		
CInv() Public		int [in] typeIn uint256& [in] hashIn
CInv() Public		std::string& [in] strType uint256& [in] hashIn
Const GetCommand() char*		

Method	Notes	Parameters
Public		
IsKnownType() bool		
Public		
operator<() bool		CInv& [in] a
Public		CInv& [in] b
SerializationOp() void		Stream& [in] s
Public		Operation [in] ser_action
		int [in] nType
		int [in] nVersion
ToString() std::string		
Public		

CKey

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {A81E8F34-97B1-4252-8674-B892F74E4685}

An encapsulated private key.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TestChain100Setup	Public coinbaseKey CKey	
Association Source -> Destination	Public CExtKey	Public key CKey	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
fCompressed bool Private	! Whether the public key corresponding to this private key is (to be) compressed.	<i>Default:</i>
fValid bool Private	! Whether this private key is valid. We check for correctness when modifying the key ! data, so fValid should always correspond to the actual state.	<i>Default:</i>
vch unsigned char Private Collection	! The actual byte data	<i>Default:</i>

Operations

Method	Notes	Parameters
Const begin() unsigned char*		
Public		
Static Check() bool Private		unsigned char* [in] vch
Static CheckSignatureElement() bool Public		unsigned char* [in] vch int [in] len bool [in] half
CKey() Public		
CKey() Public		CKey& [in] secret
~CKey() Public		
Derive() bool Public		CKey& [in] keyChild ChainCode& [in] ccChild unsigned int [in] nChild

Method	Notes	Parameters
		ChainCode& [in] cc
Const end() unsigned char*		
Public		
GetPrivKey() CPrivateKey		
Public		
GetPubKey() CPubKey		
Public		
IsCompressed() bool		
Public		
IsValid() bool		
Public		
Load() bool		CPrivateKey& [in] privkey
Public		CPubKey& [in] vchPubKey
		bool [in] fSkipCheck
MakeNewKey() void		bool [in] fCompressed
Public		
operator==() bool		CKey& [in] a
Public		CKey& [in] b
Set() void		T [in] pbegin
Public		T [in] pend
		bool [in] fCompressedIn
SetPrivKey() bool		CPrivateKey& [in] vchPrivKey
Public		bool [in] fCompressed
Sign() bool		uint256& [in] hash
Public		std::vector<unsigned char>& [in] vchSig
		uint32_t [in] test_case
SignCompact() bool		uint256& [in] hash
Public		std::vector<unsigned char>& [in] vchSig

Method	Notes	Parameters
size() unsigned int Public		
VerifyPubKey() bool Public		CPubKey& [in] vchPubKey

CKeyID

Type: **Class** **uint160**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {2FE3134C-01B0-49bf-B35A-497D2231DA2B}

A reference to a CKey: the Hash160 of its serialized public key

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CKeyID	Public uint160	

Operations

Method	Notes	Parameters
CKeyID() Public		
CKeyID() Public		uint160& [in]

CKeyMetadata

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {F44BB3C4-979F-41df-9532-E95A919AFCDB}

Custom Properties

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CKeyMetadata	Public nCreateTime int64_t	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
CURRENT_VERSION int Public Static Const		<i>Default:</i> 1
nCreateTime int64_t Public	0 means unknown	<i>Default:</i>
nVersion int Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CKeyMetadata() Public		
CKeyMetadata()		int64_t [in] nCreateTime_

Method	Notes	Parameters
Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		

CKeyPool

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {AC70263A-7F9D-4c62-A905-2BFA0C3A1297}

A key pool entry

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CKeyPool	Public nTime int64_t	
Association Source -> Destination	Public CKeyPool	Public vchPubKey CPubKey	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nTime int64_t Public		<i>Default:</i>
vchPubKey CPubKey Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CKeyPool() Public		
CKeyPool() Public		CPubKey& [in] vchPubKeyIn
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion

CKeyStore**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {260B165B-D3A2-4101-9278-A224E705CB6E}

A virtual base class for key stores

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public BaseSignatureCreator	Protected keystore CKeyStore	
Generalization Source -> Destination	Public CBasicKeyStore	Public CKeyStore	
Association Source -> Destination	Public CAffectedKeysVisitor	Private keystore CKeyStore	

Attributes

Attribute	Notes	Constraints and tags
cs_KeyStore CCriticalSection Protected		<i>Default:</i> [mutable = true]

Operations

Method	Notes	Parameters
Pure AddCScript() bool Public		CScript& [in] redeemScript
AddKey() bool Public		CKey& [in] key
Pure AddKeyPubKey() bool Public		CKey& [in] key CPubKey& [in] pubkey
Pure AddWatchOnly() bool Public		CScript& [in] dest
~CKeyStore() Public		
Pure GetCScript() bool Public		CScriptID& [in] hash CScript& [in] redeemScriptOut
Pure GetKey() bool Public		CKeyID& [in] address CKey& [in] keyOut
Pure GetKeys() void Public		std::set<CKeyID>& [in] setAddress

Method	Notes	Parameters
Pure GetPubKey() bool Public		CKeyID& [in] address CPubKey& [in] vchPubKeyOut
Pure HaveCScript() bool Public		CScriptID& [in] hash
Pure HasKey() bool Public		CKeyID& [in] address
Pure HaveWatchOnly() bool Public		CScript& [in] dest
Pure HaveWatchOnly() bool Public		
Pure RemoveWatchOnly() bool Public		CScript& [in] dest

CKeyingMaterial

Type: **Class std::vector<unsigned char, secure_allocator<unsigned char> >**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model *Keywords:*
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 G UID: {E5AF29F7-E423-4df0-9EB2-0DADEF94F6E0}

Custom Properties

- isActive = False

CKeyingMaterial

Type: **Class std::vector<unsigned char, secure_allocator<unsigned char> >**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model *Keywords:*
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 G UID: {09CEA282-FD36-4162-87AD-6F1E886559B0}

Custom Properties

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCryptoKeyStore	Private vMasterKey CKeyingMaterial	

CLevelDBBatch**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {AE5989ED-F31F-4e59-9E86-CD89FADF4DB2}

Batch of changes queued to be written to a CLevelDBWrapper

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CLevelDBBatch	Private batch WriteBatch	

Attributes

Attribute	Notes	Constraints and tags
batch leveldb::WriteBatch Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Erase() void Public		K& [in] key
Write() void		K& [in] key

Method	Notes	Parameters
Public		V& [in] value

CLevelDBWrapper

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {45FA746D-9A9F-4e04-9D63-8FE509C3CB7E}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CLevelDBWrapper	Private iteroptions ReadOptions	
Association Source -> Destination	Public CLevelDBWrapper	Private options Options	
Association Source -> Destination	Public CLevelDBWrapper	Private pdb DB	
Association Source -> Destination	Public CLevelDBWrapper	Private penv Env	
Association Source -> Destination	Public CLevelDBWrapper	Private readoptions ReadOptions	
Association Source -> Destination	Public CLevelDBWrapper	Private syncoptions WriteOptions	
Association Source -> Destination	Public CLevelDBWrapper	Private writeoptions WriteOptions	
Association Source -> Destination	Public CCoinsViewDB	Protected db CLevelDBWrapper	
Generalization Source -> Destination	Public CBlockTreeDB	Public CLevelDBWrapper	

Attributes

Attribute	Notes	Constraints and tags
iteroptions leveldb::ReadOptions Private	! options used when iterating over values of the database	<i>Default:</i>
options leveldb::Options Private	! database options used	<i>Default:</i>
pdb leveldb::DB*	! the database itself	<i>Default:</i>
penv leveldb::Env*	! custom environment this database is using (may be NULL in case of default environment)	<i>Default:</i>
readoptions leveldb::ReadOptions Private	! options used when reading from the database	<i>Default:</i>
syncoptions leveldb::WriteOptions Private	! options used when sync writing to the database	<i>Default:</i>

Attribute	Notes	Constraints and tags
writeoptions leveldb::WriteOptions Private	! options used when writing to the database	<i>Default:</i>

Operations

Method	Notes	Parameters
CLevelDBWrapper() Public		boost::filesystem::path& [in] path size_t [in] nCacheSize bool [in] fMemory bool [in] fWipe
~CLevelDBWrapper() Public		
Erase() bool Public		K& [in] key bool [in] fSync
Exists() bool Public		K& [in] key
Flush() bool Public		
NewIterator() leveldb::Iterator* Public		
Read() bool Public		K& [in] key V& [in] value
Sync() bool Public		
Write() bool Public		K& [in] key V& [in] value bool [in] fSync
WriteBatch() bool Public		CLevelDBBatch& [in] batch bool [in] fSync

Method	Notes	Parameters

CMainParams

Type: [Class](#) [CChainParams](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords](#):
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {BCD983F5-30C3-448a-A6C4-CAB3282445E1}

What makes a good checkpoint block? + Is surrounded by blocks with reasonable timestamps (no blocks before with a timestamp after, none after with timestamp before) + Contains no strange transactions

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CMainParams	Public CChainParams	

Operations

Method	Notes	Parameters
CMainParams() Public		

CMainSignals

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords](#):
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {AB9437A3-6463-441c-AE57-373E172D69D7}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
BlockChecked boost::signals2::signal<void (const CBlock&, const CValidationState&)> Public	Notifies listeners of a block validation result	<i>Default:</i>
BlockFound boost::signals2::signal<void (const uint256 &)> Public	Notifies listeners that a block has been successfully mined	<i>Default:</i>
Broadcast boost::signals2::signal<void (int64_t nBestBlockTime)> Public	Tells listeners to broadcast their data.	<i>Default:</i>
Inventory boost::signals2::signal<void (const uint256 &)> Public	Notifies listeners about an inventory item being seen on the network.	<i>Default:</i>
ScriptForMining boost::signals2::signal<void (boost::shared_ptr<CReserveScript>&)> Public	Notifies listeners that a key for mining is required (coinbase)	<i>Default:</i>

Attribute	Notes	Constraints and tags
SetBestChain boost::signals2::signal<void (const CBlockLocator &)> Public	Notifies listeners of a new active block chain.	<i>Default:</i>
SyncTransaction boost::signals2::signal<void (const CTransaction &, const CBlock *)> Public	Notifies listeners of updated transaction data (transaction, and optionally the block it is found in).	<i>Default:</i>
UpdatedTransaction boost::signals2::signal<void (const uint256 &)> Public	Notifies listeners of an updated transaction without new data (for now: a coinbase potentially becoming visible).	<i>Default:</i>

CMasterKey

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {3BF8EB08-A245-421c-955F-7340B97DC92C}

Master key for wallet encryption

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
nDerivationMethod unsigned int Public	! 0 = EVP_sha512() ! 1 = scrypt()	<i>Default:</i>
nDeriveIterations unsigned int Public		<i>Default:</i>
vchCryptedKey std::vector<unsigned char> Public		<i>Default:</i>
vchOtherDerivationParameters std::vector<unsigned char> Public	! Use this for more parameters to key derivation, ! such as the various parameters to scrypt	<i>Default:</i>
vchSalt std::vector<unsigned char> Public		<i>Default:</i>

Operations

Method	Notes	Parameters
--------	-------	------------

Method	Notes	Parameters
CMasterKey() Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion

CMedianFilter

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {F07906F9-EF88-4c16-BA74-3006760D6B3B}

Median filter over a stream of values. Returns the median of the last N numbers

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
nSize unsigned int Private		<i>Default:</i>
vSorted std::vector<T> Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
vValues std::vector<T> Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CMedianFilter() Public		unsigned int [in] size T [in] initial_value
input() void Public		T [in] value
median() T Public		
size() int Public		
sorted() std::vector<T> Public		

CMerkleBlock

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {C3C2F33E-C92B-45b9-9FB5-43A6FBF0B3CF}

Used to relay blocks as header + vector<merkle branch> to filtered nodes.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CMerkleBlock	Public txn CPartialMerkleTree	
Association Source -> Destination	Public CMerkleBlock	Public header CBlockHeader	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
header CBlockHeader Public	Public only for unit testing	<i>Default:</i>
txn CPartialMerkleTree Public		<i>Default:</i>
vMatchedTxn std::vector<std::pair<unsigned int, uint256> > Public	Public only for unit testing and relay testing (not relayed)	<i>Default:</i>

Operations

Method	Notes	Parameters
CMerkleBlock() Public		CBlock& [in] block CBloomFilter& [in] filter
CMerkleBlock() Public		CBlock& [in] block std::set<uint256>& [in] txids
CMerkleBlock() Public		
SerializationOp() void Public		Stream& [in] s

Method	Notes	Parameters
		Operation [in] ser_action int [in] nType int [in] nVersion

CMerkleTx

Type: **Class** [CTransaction](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {35F94305-8F41-4256-9F75-A801EC52928F}

A transaction with a merkle branch linking it to the block chain.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CMerkleTx	Public hashBlock uint256	
Generalization Source -> Destination	Public CMerkleTx	Public CTransaction	
Generalization Source -> Destination	Public CWalletTx	Public CMerkleTx	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_METHODS int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
fMerkleVerified bool Public	memory only	<i>Default:</i> [mutable = true]
hashBlock uint256 Public		<i>Default:</i>
nIndex int Public		<i>Default:</i>
vMerkleBranch std::vector<uint256> Public		<i>Default:</i>

Operations

Method	Notes	Parameters
AcceptToMemoryPool() bool Public		bool [in] fLimitFree bool [in] fRejectAbsurdFee
CMerkleTx() Public		
CMerkleTx() Public		CTransaction& [in] txIn
GetBlocksToMaturity() int Public		
GetDepthInMainChain() int Public		CBlockIndex*& [in] pindexRet
GetDepthInMainChain()		

Method	Notes	Parameters
int Public GetDepthInMainChainINTERNAL() int Private		CBlockIndex*& [in] pindexRet
Init() void Public		
IsInMainChain() bool Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetMerkleBranch() int Public		CBlock& [in] block

CMessageHeader

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {A5BD2758-7212-4c09-8F67-14CA9286157B}

Message header. (4) message start. (12) command. (4) size. (4) checksum.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CNetMessage	Public hdr CMessageHeader	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
nChecksum unsigned int Public		<i>Default:</i>
nMessageSize unsigned int Public		<i>Default:</i>
pchCommand char Public Collection		<i>Default:</i>
pchMessageStart char Public Collection		<i>Default:</i>

Operations

Method	Notes	Parameters
CMessageHeader() Public		MessageStartChars& [in] pchMessageStartIn
CMessageHeader() Public		MessageStartChars& [in] pchMessageStartIn char* [in] pszCommand

Method	Notes	Parameters
		unsigned int [in] nMessageSizeIn
GetCommand() std::string Public		
IsValid() bool Public		MessageStartChars& [in] messageStart
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion

<anonymous>

Type:

Enumeration

Status:

Proposed. Version 1.0. Phase 1.0.

Package:

Class Model

Keywords:

Detail:

Created on 8/30/2015. Last modified on 8/30/2015.

GUID:

{5FC2B59F-4188-4fa5-9440-CF519EE63F9E}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
COMMAND_SIZE Public		<i>Default:</i> 12

Attribute	Notes	Constraints and tags
MESSAGE_SIZE_SIZE Public		<i>Default:</i> sizeof(int)
CHECKSUM_SIZE Public		<i>Default:</i> sizeof(int)
MESSAGE_SIZE_OFFSET Public		<i>Default:</i> MESSAGE_START_SIZE + COMMAND_SIZE
CHECKSUM_OFFSET Public		<i>Default:</i> MESSAGE_SIZE_OFFSET + MESSAGE_SIZE_SIZE
HEADER_SIZE Public		<i>Default:</i> MESSAGE_START_SIZE + COMMAND_SIZE + MESSAGE_SIZE_SIZE + CHECKSUM_SIZE

CMutableTransaction

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {4F07BA4F-2B0A-46bf-A99D-B8FC1EFB5153}

A mutable version of CTransaction.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CMutableTransaction	Public nLockTime uint32_t	
Association Source -> Destination	Public CMutableTransaction	Public nVersion int32_t	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
nLockTime uint32_t Public		<i>Default:</i>
nVersion int32_t Public		<i>Default:</i>
vin std::vector<CTxIn> Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
vout std::vector<CTxOut> Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CMutableTransaction() Public		
CMutableTransaction() Public		CTransaction& [in] tx
GetHash() uint256 Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion

CNetAddr**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {631553A1-FE07-417a-97B7-F5610FCAF702}

IP address (IPv6, or IPv4 using mapped IPv6 range (::FFFF:0:0/96))

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CSubNet	Protected network CNetAddr	
Generalization Source -> Destination	Public CService	Public CNetAddr	

Connector	Source	Target	Notes
Association Source -> Destination	Public CAddrInfo	Private source CNetAddr	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
ip unsigned char Protected Collection	in network byte order	<i>Default:</i>

Operations

Method	Notes	Parameters
CNetAddr() Public		
CNetAddr() Public		struct in_addr& [in] ipv4Addr
CNetAddr() Public		char* [in] pszIp bool [in] fAllowLookup
CNetAddr() Public		std::string& [in] strIp bool [in] fAllowLookup
CNetAddr() Public		struct in6_addr& [in] pipv6Addr
GetByte() unsigned int Public		int [in] n
GetGroup() std::vector<unsigned char> Public	get canonical identifier of an address' group no two connections will be attempted to addresses with the same group	
GetHash() uint64_t		

Method	Notes	Parameters
Public		
GetIn6Addr() bool Public		struct in6_addr* [in] pipv6Addr
GetInAddr() bool Public		struct in_addr* [in] pipv4Addr
GetNetwork() enum Network Public		
GetReachabilityFrom() int Public	Calculates a metric for how reachable (*this) is from a given partner	CNetAddr* [in] paddrPartner
Init() void Public		
IsIPv4() bool Public		
IsIPv6() bool Public		
IsLocal() bool Public		
IsMulticast() bool Public		
IsRFC1918() bool Public		
IsRFC2544() bool Public		
IsRFC3849() bool Public		
IsRFC3927() bool Public		
IsRFC3964() bool Public		
IsRFC4193() bool Public		
IsRFC4380() bool Public		
IsRFC4843() bool Public		
IsRFC4862() bool Public		
IsRFC5737() bool Public		
IsRFC6052() bool Public		
IsRFC6145() bool Public		
IsRFC6598() bool Public		
IsRouteable() bool Public		
IsTor() bool Public		
IsValid() bool		

Method	Notes	Parameters
Public operator!=() bool		CNetAddr& [in] a CNetAddr& [in] b
Public operator<() bool		CNetAddr& [in] a CNetAddr& [in] b
Public operator==() bool		CNetAddr& [in] a CNetAddr& [in] b
Public SerializationOp() void		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
Public SetIP() void		CNetAddr& [in] ip
Public SetRaw() void		Network [in] network uint8_t* [in] data
Public SetSpecial() bool		std::string& [in] strName
Public ToString() std::string		
Public ToStringIP() std::string		

CNetCleanup

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {65BB95A3-B3E1-40d4-AF58-CAC452F2A7B1}

Custom Properties

Custom Properties

- isActive = False

Tagged Values

- typeSynonyms = instance_of_cnetcleanup.

Operations

Method	Notes	Parameters
CNetCleanup()		
Public		
~CNetCleanup()		
Public		

CNetMessage**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {083C6E57-61B6-4e0f-915B-64CB2235BFA4}**Custom Properties**

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CNetMessage	Public hdr CMessageHeader	
Association Source -> Destination	Public CNetMessage	Public hdrbuf CDataStream	
Association Source -> Destination	Public CNetMessage	Public nTime int64_t	
Association Source -> Destination	Public CNetMessage	Public vRecv CDataStream	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
hdr CMessageHeader Public	complete header	<i>Default:</i>
hdrbuf CDataStream Public	partially received header	<i>Default:</i>
in_data bool Public	parsing header (false) or data (true)	<i>Default:</i>
nDataPos unsigned int Public		<i>Default:</i>
nHdrPos unsigned int Public		<i>Default:</i>
nTime int64_t Public	time (in microseconds) of message receipt.	<i>Default:</i>

Attribute	Notes	Constraints and tags
vRecv CDataStream Public	received message data	<i>Default:</i>

Operations

Method	Notes	Parameters
CNetMessage() Public		CMessageHeader::MessageStartC hars& [in] pchMessageStartIn int [in] nTypeIn int [in] nVersionIn
complete() bool Public		
readData() int Public		char* [in] pch unsigned int [in] nBytes
readHeader() int Public		char* [in] pch unsigned int [in] nBytes
SetVersion() void Public		int [in] nVersionIn

CNoDestination

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {6B0E6016-0156-459e-87BF-A7E8AEC029AB}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
operator<() bool Public		CNoDestination& [in] a CNoDestination& [in] b
operator==() bool Public		CNoDestination& [in] a CNoDestination& [in] b

CNode**Type:** [Class](#)**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {DA736CD7-4689-4f3c-8571-D7E3BBF69838}

Information about a peer

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CNode	Public addr CAddress	
Association Source -> Destination	Public CNode	Public addrKnown CRollingBloomFilter	
Association Source -> Destination	Public CNode	Public addrLocal CService	
Association Source -> Destination	Public CNode	Public hashContinue uint256	
Association Source -> Destination	Public CNode	Public id NodeId	
Association Source -> Destination	Public CNode	Public nLastRecv int64_t	
Association Source -> Destination	Public CNode	Public nLastSend int64_t	
Association	Public	Public nPingNonceSent	

Connector	Source	Target	Notes
Source -> Destination	CNode	uint64_t	
Association Source -> Destination	Public CNode	Public nPingUsecStart int64_t	
Association Source -> Destination	Public CNode	Public nPingUsecTime int64_t	
Association Source -> Destination	Public CNode	Public nRecvBytes uint64_t	
Association Source -> Destination	Public CNode	Public nTimeOffset int64_t	
Association Source -> Destination	Public CNode	Public nSendBytes uint64_t	
Association Source -> Destination	Public CNode	Public nTimeConnected int64_t	
Association Source -> Destination	Public CNode	Private nTotalBytesRecv uint64_t	
Association Source -> Destination	Public CNode	Private nTotalBytesSent uint64_t	
Association Source -> Destination	Public CNode	Public pfilter CBloomFilter	
Association Source -> Destination	Public CNode	Protected setBanned banmap_t	
Association Source -> Destination	Public CNode	Public ssSend CDataStream	
Association Source -> Destination	Public CNode	Public nServices uint64_t	

Attributes

Attribute	Notes	Constraints and tags
addr CAddress Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
addrKnown CRollingBloomFilter Public		<i>Default:</i>
addrLocal CService Public		<i>Default:</i>
addrName std::string Public		<i>Default:</i>
cleanSubVer std::string Public	strSubVer is whatever byte array we read from the wire. However, this field is intended to be printed out, displayed to humans in various forms and so on. So we sanitize it and store the sanitized version in cleanSubVer. The original should be used when dealing with the network or wire types and the cleaned string used when displayed or logged.	<i>Default:</i>
cs_filter CCriticalSection Public		<i>Default:</i>
cs_inventory CCriticalSection Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
cs_setBanned CCriticalSection Protected Static		<i>Default:</i>
cs_totalBytesRecv CCriticalSection Private Static	Network usage totals	<i>Default:</i>
cs_totalBytesSent CCriticalSection Private Static		<i>Default:</i>
cs_vRecvMsg CCriticalSection Public		<i>Default:</i>
cs_vSend CCriticalSection Public		<i>Default:</i>
cs_vWhitelistedRange CCriticalSection Protected Static		<i>Default:</i>

Attribute	Notes	Constraints and tags
fClient bool Public		<i>Default:</i>
fDisconnect bool Public		<i>Default:</i>
fGetAddr bool Public		<i>Default:</i>
fInbound bool Public		<i>Default:</i>
fNetworkNode bool Public		<i>Default:</i>
fOneShot bool Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
fPingQueued bool Public	Whether a ping is requested.	<i>Default:</i>
fRelayTxes bool Public	We use fRelayTxes for two purposes - a) it allows us to not relay tx invs before receiving the peer's version message b) the peer may tell us in its version message that we should not relay tx invs until it has initialized its bloom filter.	<i>Default:</i>
fSuccessfullyConnected bool Public		<i>Default:</i>
fWhitelisted bool Public	This peer can bypass DoS banning.	<i>Default:</i>
grantOutbound CSemaphoreGrant Public		<i>Default:</i>
hashContinue uint256 Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
hSocket SOCKET Public		<i>Default:</i>
id NodeId Public		<i>Default:</i>
mapAskFor std::multimap<int64_t, CInv> Public		<i>Default:</i>
nLastRecv int64_t Public		<i>Default:</i>
nLastSend int64_t Public		<i>Default:</i>
nPingNonceSent uint64_t Public	Ping time measurement: The pong reply we're expecting, or 0 if no pong expected.	<i>Default:</i>

Attribute	Notes	Constraints and tags
nPingUsecStart int64_t Public	Time (in usec) the last ping was sent, or 0 if no ping was ever sent.	<i>Default:</i>
nPingUsecTime int64_t Public	Last measured round-trip time.	<i>Default:</i>
nRecvBytes uint64_t Public		<i>Default:</i>
nRecvVersion int Public		<i>Default:</i>
nRefCount int Public		<i>Default:</i>
nSendBytes uint64_t Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nSendOffset size_t Public	offset inside the first vSendMsg already sent	<i>Default:</i>
nSendSize size_t Public	total size of all vSendMsg entries	<i>Default:</i>
nServices uint64_t Public	socket	<i>Default:</i>
nStartingHeight int Public		<i>Default:</i>
nTimeConnected int64_t Public		<i>Default:</i>
nTimeOffset int64_t Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nTotalBytesRecv uint64_t Private Static		<i>Default:</i>
nTotalBytesSent uint64_t Private Static		<i>Default:</i>
nVersion int Public		<i>Default:</i>
pfilter CBloomFilter* Public		<i>Default:</i>
setBanned banmap_t Protected Static	Denial-of-service detection/prevention Key is IP address, value is banned-until-time	<i>Default:</i>
setBannedIsDirty bool Protected Static		<i>Default:</i>

Attribute	Notes	Constraints and tags
setInventoryKnown mruset<CInv> Public	inventory based relay	<i>Default:</i>
setKnown std::set<uint256> Public		<i>Default:</i>
ssSend CDataStream Public		<i>Default:</i>
strSubVer std::string Public	strSubVer is whatever byte array we read from the wire. However, this field is intended to be printed out, displayed to humans in various forms and so on. So we sanitize it and store the sanitized version in cleanSubVer. The original should be used when dealing with the network or wire types and the cleaned string used when displayed or logged.	<i>Default:</i>
vAddrToSend std::vector<CAddress> Public	flood relay	<i>Default:</i>
vInventoryToSend std::vector<CInv> Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
vRecvGetData std::deque<CInv> Public		<i>Default:</i>
vRecvMsg std::deque<CNetMessage> Public		<i>Default:</i>
vSendMsg std::deque<CSerializeData > Public		<i>Default:</i>
vWhitelistedRange std::vector<CSubNet> Protected Static	Whitelisted ranges. Any node connecting from these is automatically whitelisted (as well as those connecting to whitelisted binds).	<i>Default:</i>

Operations

Method	Notes	Parameters
AbortMessage() void Public		
AddAddressKnown() void Public		CAddress& [in] addr
AddInventoryKnown() void Public		CInv& [in] inv
AddRef() CNode* Public		
Static AddWhitelistedRange() void Public		CSubNet& [in] subnet
AskFor() void Public		CInv& [in] inv

Method	Notes	Parameters
Static Ban() void Public		CNetAddr& [in] ip BanReason& [in] banReason int64_t [in] bantimeoffset bool [in] sinceUnixEpoch
Static Ban() void Public		CSubNet& [in] subNet BanReason& [in] banReason int64_t [in] bantimeoffset bool [in] sinceUnixEpoch
Static BannedSetIsDirty() bool Public		
BeginMessage() void Public		char* [in] pszCommand
Static ClearBanned() void Public		
CloseSocketDisconnect() void Public		
CNode() Public		SOCKET [in] hSocketIn CAddress& [in] addrIn std::string& [in] addrNameIn bool [in] fInboundIn
~CNode() Public		
CNode() Private		CNode& [in] prm1
copyStats() void Public		CNodeStats& [in] stats
EndMessage() void Public		
EXCLUSIVE_LOCK_F UNCTION() Public		cs_vSend [in] prm1
Fuzz() void Protected		int [in] nChance

Method	Notes	Parameters
Static GetBanned() void Public		banmap_t& [in] banmap
GetId() NodeId Public		
GetRefCount() int Public		
Static GetTotalBytesRecv() uint64_t Public		
Static GetTotalBytesSent() uint64_t Public		
GetTotalRecvSize() unsigned int Public		
Static IsBanned() bool Public		CNetAddr [in] ip
Static IsBanned() bool Public		CSubNet [in] subnet
Static IsWhitelistedRange() bool Public		CNetAddr& [in] ip
operator=() void Private		CNode& [in] prm1
PushAddress() void Public		CAddress& [in] addr
PushInventory() void Public		CInv& [in] inv
PushMessage() void Public		char* [in] pszCommand
PushMessage() void Public		char* [in] pszCommand T1& [in] a1
PushMessage() void Public		char* [in] pszCommand T1& [in] a1 T2& [in] a2
PushMessage() void Public		char* [in] pszCommand T1& [in] a1

Method	Notes	Parameters
		T2& [in] a2 T3& [in] a3
PushMessage() void Public		char* [in] pszCommand T1& [in] a1 T2& [in] a2 T3& [in] a3 T4& [in] a4
PushMessage() void Public		char* [in] pszCommand T1& [in] a1 T2& [in] a2 T3& [in] a3 T4& [in] a4 T5& [in] a5
PushMessage() void Public		char* [in] pszCommand T1& [in] a1 T2& [in] a2 T3& [in] a3 T4& [in] a4 T5& [in] a5 T6& [in] a6
PushMessage() void Public		char* [in] pszCommand T1& [in] a1 T2& [in] a2 T3& [in] a3 T4& [in] a4

Method	Notes	Parameters
		T5& [in] a5 T6& [in] a6 T7& [in] a7
PushMessage() void Public		char* [in] pszCommand T1& [in] a1 T2& [in] a2 T3& [in] a3 T4& [in] a4 T5& [in] a5 T6& [in] a6 T7& [in] a7 T8& [in] a8
PushMessage() void Public		char* [in] pszCommand T1& [in] a1 T2& [in] a2 T3& [in] a3 T4& [in] a4 T5& [in] a5 T6& [in] a6 T7& [in] a7 T8& [in] a8 T9& [in] a9
PushVersion() void Public		
ReceiveMsgBytes() bool Public	requires LOCK(cs_vRecvMsg)	char* [in] pch unsigned int [in] nBytes
Static RecordBytesRecv()		uint64_t [in] bytes

Method	Notes	Parameters
void Public		
Static RecordBytesSent() void Public		uint64_t [in] bytes
Release() void Public		
Static SetBanned() void Public		banmap_t& [in] banmap
Static SetBannedSetDirty() void Public		bool [in] dirty
SetRecvVersion() void Public		int [in] nVersionIn
Static SweepBanned() void Public		
Static Unban() bool Public		CNetAddr& [in] ip
Static Unban() bool Public		CSubNet& [in] ip
UNLOCK_FUNCTION() Public		cs_vSend [in] prm1
UNLOCK_FUNCTION() Public		cs_vSend [in] prm1

CNodeSignals

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {1CE9C060-E31D-4fd0-84CC-73003BD7B91B}

Signals for message handling

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
FinalizeNode boost::signals2::signal<void (NodeId)> Public		<i>Default:</i>
GetHeight boost::signals2::signal<int ()> Public		<i>Default:</i>
InitializeNode boost::signals2::signal<void (NodeId, const CNode*)> Public		<i>Default:</i>
ProcessMessages boost::signals2::signal<bool (CNode*, CombinerAll)> Public		<i>Default:</i>
SendMessages boost::signals2::signal<bool (CNode*, bool), CombinerAll> Public		<i>Default:</i>

CNodeStateStats

Type: **Class**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {D3766B07-8414-4fda-8014-F172A1AE248F}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
nCommonHeight int Public		<i>Default:</i>
nMisbehavior int Public		<i>Default:</i>
nSyncHeight int Public		<i>Default:</i>
vHeightInFlight std::vector<int> Public		<i>Default:</i>

CNodeStats

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {9406575C-EF25-4ced-8C09-4F738D1338AA}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CNodeStats	Public nLastRecv int64_t	
Association Source -> Destination	Public CNodeStats	Public nLastSend int64_t	
Association Source -> Destination	Public CNodeStats	Public nodeid NodeId	
Association Source -> Destination	Public CNodeStats	Public nRecvBytes uint64_t	
Association Source -> Destination	Public CNodeStats	Public nSendBytes uint64_t	
Association Source -> Destination	Public CNodeStats	Public nServices uint64_t	
Association Source -> Destination	Public CNodeStats	Public nTimeOffset int64_t	
Association Source -> Destination	Public CNodeStats	Public nTimeConnected int64_t	

Attributes

Attribute	Notes	Constraints and tags
addrLocal std::string Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
addrName std::string Public		<i>Default:</i>
cleanSubVer std::string Public		<i>Default:</i>
dPingTime double Public		<i>Default:</i>
dPingWait double Public		<i>Default:</i>
fInbound bool Public		<i>Default:</i>
fWhitelisted bool Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nLastRecv int64_t Public		<i>Default:</i>
nLastSend int64_t Public		<i>Default:</i>
nodeid NodeId Public		<i>Default:</i>
nRecvBytes uint64_t Public		<i>Default:</i>
nSendBytes uint64_t Public		<i>Default:</i>
nServices uint64_t Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nStartingHeight int Public		<i>Default:</i>
nTimeConnected int64_t Public		<i>Default:</i>
nTimeOffset int64_t Public		<i>Default:</i>
nVersion int Public		<i>Default:</i>

COrphan

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {5BE8AA4B-BDB6-422e-917F-B6D1C81D8849}

Unconfirmed transactions in the memory pool often depend on other transactions in the memory pool. When we select transactions from the pool, we select by highest priority or fee rate, so we might consider transactions that depend on transactions that aren't yet in the block. The COrphan class keeps track of these 'temporary orphans' while CreateBlock is figuring out which transactions to include.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public COrphan	Public ptx CTransaction	
Association Source -> Destination	Public COrphan	Public feeRate CFeeRate	

Attributes

Attribute	Notes	Constraints and tags
dPriority double Public		<i>Default:</i>
feeRate CFeeRate Public		<i>Default:</i>
ptx CTransaction* Public Const		<i>Default:</i>
setDependsOn set<uint256> Public		<i>Default:</i>

Operations

Method	Notes	Parameters
COrphan() Public		CTransaction* [in] ptxIn

COrphanTx

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {4F7D4436-05C2-4544-A2A5-55372FAAE633}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public COrphanTx	Public fromPeer NodeId	
Association Source -> Destination	Public COrphanTx	Public tx CTransaction	

Attributes

Attribute	Notes	Constraints and tags
fromPeer NodeId Public		<i>Default:</i>
tx CTransaction Public		<i>Default:</i>

COutPoint

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {432F8120-5400-4c23-93EC-E3A344837FE6}

An outpoint - a combination of a transaction hash and an index n into its vout

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public COutPoint	Public hash uint256	
Association Source -> Destination	Public COutPoint	Public n uint32_t	
Association Source -> Destination	Public CTxIn	Public prevout COutPoint	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_METHODS int Public		<i>Default:</i>
hash uint256 Public		<i>Default:</i>
n uint32_t Public		<i>Default:</i>

Operations

Method	Notes	Parameters
COutPoint() Public		
COutPoint()		uint256 [in] hashIn

Method	Notes	Parameters
Public		uint32_t [in] nIn
IsNull() bool Public		
operator!=() bool Public		COutPoint& [in] a COutPoint& [in] b
operator<() bool Public		COutPoint& [in] a COutPoint& [in] b
operator==() bool Public		COutPoint& [in] a COutPoint& [in] b
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		
ToString() std::string Public		

COutput

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {98BF2116-5AFE-48bf-98F7-DC9FECC15DB2}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public COutput	Public tx CWalletTx	

Attributes

Attribute	Notes	Constraints and tags
fSpendable bool Public		<i>Default:</i>
i int Public		<i>Default:</i>
nDepth int Public		<i>Default:</i>
tx CWalletTx* Public Const		<i>Default:</i>

Operations

Method	Notes	Parameters
COutput() Public		CWalletTx* [in] txIn int [in] iIn int [in] nDepthIn bool [in] fSpendableIn
ToString() std::string		

Method	Notes	Parameters
Public		

COutputEntry

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {9A173362-3B3A-41c7-8DF2-8A0E25F60D0C}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public COutputEntry	Public amount CAmount	
Association Source -> Destination	Public COutputEntry	Public destination CTxDestination	

Attributes

Attribute	Notes	Constraints and tags
amount CAmount Public		<i>Default:</i>
destination CTxDestination Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
vout int Public		<i>Default:</i>

CPartialMerkleTree

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model

Keywords:

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {F7ADF5CB-ABC9-44dc-8E3F-3752CE302C95}

Data structure that represents a partial merkle tree. It represents a subset of the txid's of a known block, in a way that allows recovery of the list of txid's and the merkle root, in an authenticated way. The encoding works as follows: we traverse the tree in depth-first order, storing a bit for each traversed node, signifying whether the node is the parent of at least one matched leaf txid (or a matched txid itself). In case we are at the leaf level, or this bit is 0, its merkle node hash is stored, and its children are not explored further. Otherwise, no hash is stored, but we recurse into both (or the only) child branch. During decoding, the same depth-first traversal is performed, consuming bits and hashes as they were written during encoding. The serialization is fixed and provides a hard guarantee about the encoded size: SIZE <= 10 + ceil(32.25*N) Where N represents the number of leaf nodes of the partial tree. N itself is bounded by: N <= total_transactions N <= 1 + matched_transactions*tree_height The serialization format: - uint32 total_transactions (4 bytes) - varint number of hashes (1-3 bytes) - uint256[] hashes in depth-first order (<= 32*N bytes) - varint number of bytes of flag bits (1-3 bytes) - byte[] flag bits, packed per 8 in a byte, least significant bit first (<= 2*N-1 bits) The size constraints follow from this.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CMerkleBlock	Public txn CPartialMerkleTree	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public	serialization implementation	<i>Default:</i>
fBad bool Protected	flag set when encountering invalid data	<i>Default:</i>
nTransactions unsigned int Protected	the total number of transactions in the block	<i>Default:</i>
vBits std::vector<bool> Protected	node-is-parent-of-matched-txid bits	<i>Default:</i>
vHash std::vector<uint256> Protected	txids and internal hashes	<i>Default:</i>

Operations

Method	Notes	Parameters
CalcHash() uint256 Protected		int [in] height unsigned int [in] pos std::vector<uint256>& [in] vTxid
CalcTreeWidth() unsigned int		int [in] height

Method	Notes	Parameters
Protected		
CPartialMerkleTree() Public		std::vector<uint256>& [in] vTxid std::vector<bool>& [in] vMatch
CPartialMerkleTree() Public		
ExtractMatches() uint256 Public		std::vector<uint256>& [in] vMatch
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
TraverseAndBuild() void Protected		int [in] height unsigned int [in] pos std::vector<uint256>& [in] vTxid std::vector<bool>& [in] vMatch
TraverseAndExtract() uint256 Protected		int [in] height unsigned int [in] pos unsigned int& [in] nBitsUsed unsigned int& [in] nHashUsed std::vector<uint256>& [in] vMatch

CPrivKey

Type: **Class** **std::vector<unsigned char, secure_allocator<unsigned char> >**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {C0B4075D-E242-48ad-BD52-111ABE813CEE}

secure_allocator is defined in allocators.h CPrivKey is a serialized private key, with all parameters included (279 bytes)

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CWalletKey	Public vchPrivKey CPrivateKey	

CPubKey

Type: Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {6112183A-9A10-46d7-AC15-47D12C48D18A}

An encapsulated public key.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CExtPubKey	Public pubkey CPubKey	
Association Source -> Destination	Public CReserveKey	Protected vchPubKey CPubKey	
Association Source -> Destination	Public CAccount	Public vchPubKey CPubKey	
Association Source -> Destination	Public CKeyPool	Public vchPubKey CPubKey	
Association Source -> Destination	Public CWallet	Public vchDefaultKey CPubKey	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
vch unsigned char Private Collection	Just store the serialized data. Its length can very cheaply be computed from the first byte.	<i>Default:</i>

Operations

Method	Notes	Parameters
Const begin() unsigned char* Public		
CPubKey() Public		
CPubKey() Public		T [in] pbegin T [in] pend
CPubKey() Public		std::vector<unsigned char>& [in] vch
Decompress() bool Public		
Derive() bool Public		CPubKey& [in] pubkeyChild ChainCode& [in] ccChild unsigned int [in] nChild ChainCode& [in] cc
Const end() unsigned char* Public		
GetHash() uint256 Public		
GetID() CKeyID Public		
Static GetLen() unsigned int Private		unsigned char [in] chHeader
GetSerializeSize() unsigned int Public		int [in] nType int [in] nVersion
Invalidate() void Private		

Method	Notes	Parameters
IsCompressed() bool Public		
IsFullyValid() bool Public		
IsValid() bool Public		
operator!=() bool Public		CPubKey& [in] a CPubKey& [in] b
Const operator[]() unsigned char& Public		unsigned int [in] pos
operator<() bool Public		CPubKey& [in] a CPubKey& [in] b
operator==() bool Public		CPubKey& [in] a CPubKey& [in] b
RecoverCompact() bool Public		uint256& [in] hash std::vector<unsigned char>& [in] vchSig
Serialize() void Public		Stream& [in] s int [in] nType int [in] nVersion
Set() void Public		T [in] pbegin T [in] pend
size() unsigned int Public		
Unserialize() void Public		Stream& [in] s int [in] nType int [in] nVersion
Verify() bool Public		uint256& [in] hash std::vector<unsigned char>& [in] vchSig

Method	Notes	Parameters

CRIPEMD160

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {B51EA0FA-C173-4602-9B48-241810FBF670}

A hasher class for RIPEMD-160.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CRIPEMD160	Private s uint32_t	

Attributes

Attribute	Notes	Constraints and tags
buf unsigned char Private Collection		<i>Default:</i>
bytes size_t Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
OUTPUT_SIZE size_t Public Static Const		<i>Default:</i> 20
s uint32_t Private Collection		<i>Default:</i>

Operations

Method	Notes	Parameters
CRIPEMD160() Public	RIPEMD160	
Finalize() void Public		unsigned char [in] hash[OUTPUT_SIZE]
Reset() CRIPEMD160& Public		
Write() CRIPEMD160& Public		unsigned char* [in] data size_t [in] len

CRPCCommand

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {2FBBA06E-28F3-4702-9F00-7EA611FF9421}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
actor rpcfn_type Public		<i>Default:</i>
category std::string Public		<i>Default:</i>
name std::string Public		<i>Default:</i>
okSafeMode bool Public		<i>Default:</i>

CRPCConvertParam

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {05970FF8-1596-475f-B194-9368C50D4A22}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
methodName std::string Public	! method whose params want conversion	<i>Default:</i>
paramIdx int Public	! 0-based idx of param to convert	<i>Default:</i>

CRPCCConvertTable

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {221BB80A-8AC6-47fa-AF3E-9CC1450305D1}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
members std::set<std::pair<std::string, int> > Private		<i>Default:</i>

Operations

Method	Notes	Parameters
convert() bool Public		std::string& [in] method int [in] idx

Method	Notes	Parameters
CRPCConvertTable() Public		

CRPCTable

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {F827B5C7-0318-459d-B31F-F29FC23C3434}

Bitcoin RPC command dispatcher.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
mapCommands std::map<std::string, const CRPCCommand*> Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CRPCTable() Public		
execute() UniValue Public		std::string& [in] method UniValue& [in] params
help() std::string Public		std::string& [in] name
Const operator[]() CRPCCommand* Public		std::string& [in] name

CRecipient

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {CB107ABF-EC56-4695-9D6A-513259806520}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CRecipient	Public nAmount CAmount	
Association Source -> Destination	Public CRecipient	Public scriptPubKey CScript	

Attributes

Attribute	Notes	Constraints and tags
fSubtractFeeFromAmount bool Public		<i>Default:</i>
nAmount CAmount Public		<i>Default:</i>
scriptPubKey CScript Public		<i>Default:</i>

CRegTestParams

Type:	Class	CChainParams
Status:	Proposed.	Version 1.0. Phase 1.0.
Package:	Class Model	<i>Keywords:</i>
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>	
GUID:	{F0C9D61F-3E6D-497a-9B1A-F531FEBECC0A}	

Regression test

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CRegTestParams	Public CChainParams	

Operations

Method	Notes	Parameters
CRegTestParams() Public		

CReserveKey

Type:	Class	CReserveScript
Status:	Proposed.	Version 1.0. Phase 1.0.
Package:	Class Model	<i>Keywords:</i>
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>	
GUID:	{E15DA6A4-DCEE-4119-910A-78000E8C29BE}	

A key allocated from the key pool.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public WalletModelTransactio n	Private keyChange CReserveKey	
Association Source -> Destination	Public CReserveKey	Protected pwallet CWallet	

Connector	Source	Target	Notes
Association Source -> Destination	Public CReserveKey	Protected vchPubKey CPubKey	
Generalization Source -> Destination	Public CReserveKey	Public CReserveScript	
Association Source -> Destination	Public CReserveKey	Protected nIndex int64_t	

Attributes

Attribute	Notes	Constraints and tags
nIndex int64_t Protected		<i>Default:</i>
pwallet CWallet* Protected		<i>Default:</i>
vchPubKey CPubKey Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
CReserveKey() Public		CWallet* [in] pwalletIn
~CReserveKey() Public		
GetReservedKey() bool Public		CPubKey& [in] pubkey
KeepKey() void Public		
KeepScript() void Public		

Method	Notes	Parameters
ReturnKey() void Public		

CReserveScript

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {88F18C92-9C00-4a53-82EF-D1323934BEC8}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CReserveScript	Public reserveScript CScript	
Generalization Source -> Destination	Public CReserveKey	Public CReserveScript	

Attributes

Attribute	Notes	Constraints and tags
reserveScript CScript Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CReserveScript() Public		
~CReserveScript() Public		
KeepScript() void Public		

CRollingBloomFilter

Type:	Class
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model
Detail:	<i>Keywords:</i> Created on 8/30/2015. Last modified on 8/30/2015.
GUID:	{6EDD4619-7E88-4599-A3ED-0B0DDB5F08CF}

RollingBloomFilter is a probabilistic "keep track of most recently inserted" set. Construct it with the number of items to keep track of, and a false-positive rate. Unlike CBloomFilter, by default nTweak is set to a cryptographically secure random value for you. Similarly rather than clear() the method reset() is provided, which also changes nTweak to decrease the impact of false-positives. contains(item) will always return true if item was one of the last N things insert()'ed ... but may also return true for items that were not inserted.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CNode	Public addrKnown CRollingBloomFilter	
Association Source -> Destination	Public CRollingBloomFilter	Private b1 CBloomFilter	
Association Source -> Destination	Public CRollingBloomFilter	Private b2 CBloomFilter	

Attributes

Attribute	Notes	Constraints and tags
b1 CBloomFilter Private		<i>Default:</i>
b2 CBloomFilter Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
nBloomSize unsigned int Private		<i>Default:</i>
nInsertions unsigned int Private		<i>Default:</i>

Operations

Method	Notes	Parameters
contains() bool Public		std::vector<unsigned char>& [in] vKey
contains() bool Public		uint256& [in] hash
CRollingBloomFilter() Public		unsigned int [in] nElements double [in] nFPRate
insert() void Public		std::vector<unsigned char>& [in] vKey
insert() void Public		uint256& [in] hash
reset() void Public		

CSHA1

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {9C97633B-7854-400d-875F-2D99AB7A5481}

A hasher class for SHA1.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CSHA1	Private s uint32_t	

Attributes

Attribute	Notes	Constraints and tags
buf unsigned char Private Collection		<i>Default:</i>
bytes size_t Private		<i>Default:</i>
OUTPUT_SIZE size_t Public Static Const		<i>Default:</i> 20
s uint32_t Private Collection		<i>Default:</i>

Operations

Method	Notes	Parameters
CSHA1() Public	SHA1	

Method	Notes	Parameters
Finalize() void Public		unsigned char [in] hash[OUTPUT_SIZE]
Reset() CSHA1& Public		
Write() CSHA1& Public		unsigned char* [in] data size_t [in] len

CSHA256

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A5DBEE45-D56B-47f7-A6D7-6CECC5620998}

A hasher class for SHA-256.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CHMAC_SHA256	Private inner CSHA256	
Association Source -> Destination	Public CHMAC_SHA256	Private outer CSHA256	
Association Source -> Destination	Public CSHA256	Private s uint32_t	
Association Source -> Destination	Public CHash160	Private sha CSHA256	
Association Source -> Destination	Public CHash256	Private sha CSHA256	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
buf unsigned char Private Collection		<i>Default:</i>
bytes size_t Private		<i>Default:</i>
OUTPUT_SIZE size_t Public Static Const		<i>Default:</i> 32
s uint32_t Private Collection		<i>Default:</i>

Operations

Method	Notes	Parameters
CSHA256() Public	SHA-256	
Finalize() void Public		unsigned char [in] hash[OUTPUT_SIZE]
Reset() CSHA256& Public		
Write() CSHA256& Public		unsigned char* [in] data size_t [in] len

CSHA512

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {71C020DB-B172-4ade-B6BF-12A9A63CE21E}

A hasher class for SHA-512.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CHMAC_SHA512	Private inner CSHA512	
Association Source -> Destination	Public CHMAC_SHA512	Private outer CSHA512	
Association Source -> Destination	Public CSHA512	Private s uint64_t	

Attributes

Attribute	Notes	Constraints and tags
buf unsigned char Private Collection		<i>Default:</i>
bytes size_t Private		<i>Default:</i>
OUTPUT_SIZE size_t Public Static Const		<i>Default:</i> 64

Attribute	Notes	Constraints and tags
s uint64_t Private Collection		<i>Default:</i>

Operations

Method	Notes	Parameters
CSHA512()	SHA-512	
Public		
Finalize() void		unsigned char [in] hash[OUTPUT_SIZE]
Public		
Reset() CSHA512&		
Public		
Write() CSHA512&		unsigned char* [in] data
Public		size_t [in] len

CScheduler

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {BC19DF7F-BF12-4502-A705-E0519854832E}

Simple class for background tasks that should be run periodically or once "after a while". Usage: CScheduler* s = new CScheduler(); s->scheduleFromNow(doSomething, 11); // Assuming a: void doSomething() { } s->scheduleFromNow(boost::bind(Class::func, this, argument), 3); boost::thread* t = new boost::thread(boost::bind(CScheduler::serviceQueue, s)); ... then at program shutdown, clean up the thread running serviceQueue: t->interrupt(); t->join(); delete t; delete s; // Must be done after thread is interrupted/joined.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
newTaskMutex boost::mutex Private		<i>Default:</i> [mutable = true]
newTaskScheduled boost::condition_variable Private		<i>Default:</i>
nThreadsServicingQueue int Private		<i>Default:</i>
stopRequested bool Private		<i>Default:</i>
stopWhenEmpty bool Private		<i>Default:</i>
taskQueue std::multimap<boost::chrono::system_clock::time_point, Function> Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CScheduler() Public		
~CScheduler() Public		
getQueueInfo() size_t Public		boost::chrono::system_clock::time_point& [in] first boost::chrono::system_clock::time_point& [in] last
schedule() void Public		Function [in] f boost::chrono::system_clock::time_point [in] t
scheduleEvery() void Public		Function [in] f int64_t [in] deltaSeconds
scheduleFromNow() void Public		Function [in] f int64_t [in] deltaSeconds
serviceQueue() void Public		
shouldStop() bool Private		
stop() void Public		bool [in] drain

Function

Type: **Class** [boost::function<void\(void\)>](#)
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 GUID: {FA4AB001-24B3-4fb0-9942-B05C16DE71A4}

Custom Properties

- isActive = False

CScript

Type: **Class** **`std::vector`**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {7D5C9EAC-5DB3-4e07-98F5-3891D08ABFB8}

Serialized script, used inside transaction inputs and outputs

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTxOut	Public scriptPubKey CScript	
Association Source -> Destination	Public CTxIn	Public scriptSig CScript	
Association Source -> Destination	Public CReserveScript	Public reserveScript CScript	
Association Source -> Destination	Public CTransactionSignature Serializer	Private scriptCode CScript	
Association Source -> Destination	Public CScriptVisitor	Private script CScript	
Association Source -> Destination	Public CScriptCheck	Private scriptPubKey CScript	
Association Source -> Destination	Public CRecipient	Public scriptPubKey CScript	
Association Source -> Destination	Public CScriptCompressor	Private script CScript	

Operations

Method	Notes	Parameters
clear() void Public		
CScript() Public		
CScript() Public		CScript& [in] b
CScript()		const_iterator [in] pbegin

Method	Notes	Parameters
Public		const_iterator [in] pend
CScript() Public		unsigned char* [in] pbegin unsigned char* [in] pend
CScript() Public		int64_t [in] b
CScript() Public		opcotype [in] b
CScript() Public		CScriptNum& [in] b
CScript() Public		std::vector<unsigned char>& [in] b
Static DecodeOP_N() int Public		opcotype [in] opcode
Static EncodeOP_N() opcotype Public		int [in] n
Find() int Public		opcotype [in] op
FindAndDelete() int Public		CScript& [in] b
GetOp() bool Public		iterator& [in] pc opcotype& [in] opcodeRet std::vector<unsigned char>& [in] vchRet
GetOp() bool Public		iterator& [in] pc opcotype& [in] opcodeRet
GetOp() bool Public		const_iterator& [in] pc opcotype& [in] opcodeRet std::vector<unsigned char>& [in] vchRet

Method	Notes	Parameters
GetOp() bool Public		const_iterator& [in] pc opcodetype& [in] opcodeRet
GetOp2() bool Public		const_iterator& [in] pc opcodetype& [in] opcodeRet std::vector<unsigned char>*& [in] pvchRet
GetSigOpCount() unsigned int Public		bool [in] fAccurate
GetSigOpCount() unsigned int Public		CScript& [in] scriptSig
IsPayToScriptHash() bool Public		
IsPushOnly() bool Public		
IsUnspendable() bool Public		
operator+() CScript Public		CScript& [in] a CScript& [in] b
operator+=() CScript& Public		CScript& [in] b
operator<<() CScript& Public		int64_t [in] b
operator<<() CScript& Public		opcodetype [in] opcode
operator<<() CScript& Public		CScriptNum& [in] b
operator<<() CScript& Public		std::vector<unsigned char>& [in] b
operator<<() CScript& Public		CScript& [in] b
push_int64() CScript& Protected		int64_t [in] n
ToString() std::string Public		

CScriptCheck

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {842E4F52-A241-4036-B11E-A8202BE9C2EA}

Closure representing one script verification Note that this stores references to the spending transaction

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CScriptCheck	Private ptxTo CTransaction	
Association Source -> Destination	Public CScriptCheck	Private scriptPubKey CScript	

Attributes

Attribute	Notes	Constraints and tags
cacheStore bool Private		<i>Default:</i>
error ScriptError Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
nFlags unsigned int Private		<i>Default:</i>
nIn unsigned int Private		<i>Default:</i>
ptxTo CTransaction* Private Const		<i>Default:</i>
scriptPubKey CScript Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CScriptCheck() Public		
CScriptCheck() Public		CCoins& [in] txFromIn CTransaction& [in] txToIn unsigned int [in] nInIn unsigned int [in] nFlagsIn bool [in] cacheIn
GetScriptError() ScriptError Public		
operator() bool		

Method	Notes	Parameters
Public		
swap() void Public		CScriptCheck& [in] check

CScriptCompressor

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {C35AE5F7-A549-4fb1-B1A9-D18D9B36DAEB}

Compact serializer for scripts. It detects common cases and encodes them much more efficiently. 3 special cases are defined: Pay to pubkey hash (encoded as 21 bytes) Pay to script hash (encoded as 21 bytes) Pay to pubkey starting with 0x02, 0x03 or 0x04 (encoded as 33 bytes) Other scripts up to 121 bytes require 1 byte + script length. Above that, scripts up to 16505 bytes require 2 bytes + script length.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CScriptCompressor	Private script CScript	

Attributes

Attribute	Notes	Constraints and tags
nSpecialScripts unsigned int Private Static Const	make this static for now (there are only 6 special scripts defined) this can potentially be extended together with a new nVersion for transactions, in which case this value becomes dependent on nVersion and nHeight of the enclosing transaction.	<i>Default:</i> 6
script CScript& Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Compress() bool Protected		std::vector<unsigned char>& [in] out
CScriptCompressor() Public		CScript& [in] scriptIn
Decompress() bool Protected		unsigned int [in] nSize std::vector<unsigned char>& [in] out
GetSerializeSize() unsigned int Public		int [in] nType int [in] nVersion
GetSpecialSize() unsigned int Protected		unsigned int [in] nSize
IsToKeyID() bool Protected		CKeyID& [in] hash
IsToPubKey() bool Protected		CPubKey& [in] pubkey
IsToScriptID() bool Protected		CScriptID& [in] hash
Serialize() void Public		Stream& [in] s int [in] nType int [in] nVersion
Unserialize() void Public		Stream& [in] s int [in] nType int [in] nVersion

CScriptID

Type: Class uint160
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {252936E1-3442-4c01-B68E-21626A76A18C}

A reference to a CScript: the Hash160 of its serialization (see script.h)

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CScriptID	Public uint160	

Operations

Method	Notes	Parameters
CScriptID() Public		
CScriptID() Public		CScript& [in] in
CScriptID() Public		uint160& [in] in

CScriptNum

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {14163F84-3A1B-4eec-B910-B60C13517D04}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CScriptNum	Private m_value int64_t	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
m_value int64_t Private		<i>Default:</i>
nDefaultMaxNumSize size_t Public Static Const		<i>Default:</i> 4

Operations

Method	Notes	Parameters
CScriptNum() Public		int64_t& [in] n
CScriptNum() Public		std::vector<unsigned char>& [in] vch bool [in] fRequireMinimal size_t [in] nMaxNumSize
getint() int Public		
getvch() std::vector<unsigned char> Public		
operator-() CScriptNum Public		int64_t& [in] rhs
operator-() CScriptNum Public		CScriptNum& [in] rhs
operator-() CScriptNum Public		
operator-=() CScriptNum& Public		CScriptNum& [in] rhs
operator-=() CScriptNum& Public		int64_t& [in] rhs
operator!=() bool Public		int64_t& [in] rhs
operator!=() bool		CScriptNum& [in] rhs

Method	Notes	Parameters
Public		
operator+() CScriptNum Public		int64_t& [in] rhs
operator+() CScriptNum Public		CScriptNum& [in] rhs
operator+=() CScriptNum& Public		CScriptNum& [in] rhs
operator+=() CScriptNum& Public		int64_t& [in] rhs
operator<() bool Public		int64_t& [in] rhs
operator<() bool Public		CScriptNum& [in] rhs
operator<=() bool Public		int64_t& [in] rhs
operator<=() bool Public		CScriptNum& [in] rhs
operator=() CScriptNum& Public		int64_t& [in] rhs
operator==() bool Public		int64_t& [in] rhs
operator==() bool Public		CScriptNum& [in] rhs
operator>() bool Public		int64_t& [in] rhs
operator>() bool Public		CScriptNum& [in] rhs
operator>=() bool Public		int64_t& [in] rhs
operator>=() bool Public		CScriptNum& [in] rhs
Static serialize() std::vector<unsigned char> Public		int64_t& [in] value
Static set_vch() int64_t Private		std::vector<unsigned char>& [in] vch

CScriptVisitor

Type: **Class** **`boost::static_visitor`**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {73945EB7-37CA-4cc6-9F45-8FD86EA34CF6}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CScriptVisitor	Private script CScript	

Attributes

Attribute	Notes	Constraints and tags
script CScript* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CScriptVisitor() Public		CScript* [in] scriptin
operator()() bool Public		CNoDestination& [in] dest
operator()() bool Public		CKeyID& [in] keyID
operator()() bool Public		CScriptID& [in] scriptID

CSerActionSerialize

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {CDA08297-71B9-4dc6-AD3D-6782F6626CB9}

Support for ADD_SERIALIZE_METHODS and READWRITE macro

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
ForRead() bool Public		

CSerActionUnserialize

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {B5E02569-1E94-41f1-A309-6C735A439F1A}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
ForRead() bool Public		

CSerializeData

Type: Class std::vector<char, zero_after_free_allocator<char>>
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {1F8E9A19-4524-4617-9FE2-C02ECD118CA9}

Byte-vector that clears its contents before deletion.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public vector_type	Public CSerializeData	

CService**Type:** Class CNetAddr**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {080CBC90-97EA-41b7-9A14-7CE2E04CFCC6}

A combination of a network address (CNetAddr) and a (TCP) port

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CNode	Public addrLocal CService	
Generalization Source -> Destination	Public CService	Public CNetAddr	
Association Source -> Destination	Public proxyType	Public proxy CService	
Generalization Source -> Destination	Public CAddress	Public CService	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_METHODS int Public		<i>Default:</i>
port unsigned short Protected	host order	<i>Default:</i>

Operations

Method	Notes	Parameters
CService()		
Public		
CService()		CNetAddr& [in] ip unsigned short [in] port
Public		
CService()		struct in_addr& [in] ipv4Addr unsigned short [in] port
Public		
CService()		struct sockaddr_in& [in] addr
Public		
CService()		char* [in] pszIpPort int [in] portDefault bool [in] fAllowLookup
Public		
CService()		char* [in] pszIpPort bool [in] fAllowLookup
Public		
CService()		std::string& [in] strIpPort int [in] portDefault bool [in] fAllowLookup
Public		

Method	Notes	Parameters
CService() Public		std::string& [in] strIpPort bool [in] fAllowLookup
CService() Public		struct in6_addr& [in] ipv6Addr unsigned short [in] port
CService() Public		struct sockaddr_in6& [in] addr
GetKey() std::vector<unsigned char> Public		
GetPort() unsigned short Public		
GetSockAddr() bool Public		struct sockaddr* [in] paddr socklen_t* [in] addrlen
Init() void Public		
operator!=() bool Public		CService& [in] a CService& [in] b
operator<() bool Public		CService& [in] a CService& [in] b
operator==() bool Public		CService& [in] a CService& [in] b
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetPort() void Public		unsigned short [in] portIn
SetSockAddr() bool Public		struct sockaddr* [in] paddr

Method	Notes	Parameters
ToString() std::string Public		
ToStringIPPort() std::string Public		
ToStringPort() std::string Public		

CSignatureCache

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {ECEF84C6-B9B4-4d1b-A03B-1D9A223DB622}

Valid signature cache, to avoid doing expensive ECDSA signature checking twice for every transaction (once when accepted into memory pool, and again when accepted into the block chain)

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
cs_sigcache boost::shared_mutex Private		<i>Default:</i>
setValid std::set<sigdata_type> Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Get() bool Public		uint256& [in] hash std::vector<unsigned char>& [in] vchSig

Method	Notes	Parameters
		CPubKey& [in] pubKey
Set() void Public		uint256& [in] hash std::vector<unsigned char>& [in] vchSig CPubKey& [in] pubKey

sigdata_type

Type: **Class** `boost::tuple<uint256, std::vector<unsigned char>, CPubKey>`

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {247D5C47-1560-4a47-A763-4CAB38C57504}

! sigdata_type is (signature hash, signature, public key):

Custom Properties

- isActive = False

CSizeComputer

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {A9088CCD-B3B3-4932-A342-18120B2A5630}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
nSize size_t Protected		<i>Default:</i>
nType int Public		<i>Default:</i>
nVersion int Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CSizeComputer() Public		int [in] nTypeIn int [in] nVersionIn
operator<<() CSizeComputer& Public		T& [in] obj
size() size_t Public		
write() CSizeComputer& Public		char* [in] psz size_t [in] nSize

CSubNet**Type:****Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package:Class Model **Keywords:****Detail:**

Created on 8/30/2015. Last modified on 8/30/2015.

GUID:

{6421995C-8BBB-4fc5-9809-93CF5E6FF375}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CSubNet	Protected netmask uint8_t	
Association Source -> Destination	Public CSubNet	Protected network CNetAddr	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
netmask uint8_t Protected Collection	Netmask, in network byte order	<i>Default:</i>
network CNetAddr Protected	Network (base) address	<i>Default:</i>
valid bool Protected	Is this value valid? (only used to signal parse errors)	<i>Default:</i>

Operations

Method	Notes	Parameters
CSubNet() Public		
CSubNet() Public		std::string& [in] strSubnet bool [in] fAllowLookup
CSubNet() Public		CNetAddr& [in] addr
IsValid() bool Public		
Match() bool Public		CNetAddr& [in] addr
operator!=() bool Public		CSubNet& [in] a CSubNet& [in] b
operator<() bool Public		CSubNet& [in] a CSubNet& [in] b
operator==() bool Public		CSubNet& [in] a CSubNet& [in] b
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
ToString() std::string Public		

CTestNetParams

Type: **Class** **CChainParams**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {C4C4898E-8604-4511-9947-53CE6C5F4337}

Testnet (v3)

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CTestNetParams	Public CChainParams	

Operations

Method	Notes	Parameters
CTestNetParams() Public		

CTransaction

Type: Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {D1615EEE-2D20-4af0-AE26-47FF05393534}

The basic transaction that is broadcasted on the network and contained in blocks. A transaction can contain multiple inputs and outputs.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTransaction	Public nLockTime uint32_t	
Association Source -> Destination	Public CTransaction	Public nVersion int32_t	
Association Source -> Destination	Public CTransaction	Public CURRENT_VERSION int32_t	
Association Source -> Destination	Public CTransaction	Private hash uint256	
Association	Public	Private txTo	

Connector	Source	Target	Notes
Source -> Destination	TransactionSignatureChecker	CTransaction	
Association Source -> Destination	Public MutableTransactionSignatureChecker	Private txTo CTransaction	
Association Source -> Destination	Public TransactionSignatureCreator	Private txTo CTransaction	
Association Source -> Destination	Public CTransactionSignatureSerializer	Private txTo CTransaction	
Association Source -> Destination	Public COrphan	Public ptx CTransaction	
Association Source -> Destination	Public COrphanTx	Public tx CTransaction	
Association Source -> Destination	Public CTxMemPoolEntry	Private tx CTransaction	
Association Source -> Destination	Public CInPoint	Public ptx CTransaction	
Association Source -> Destination	Public CScriptCheck	Private ptxTo CTransaction	
Generalization Source -> Destination	Public CMerkleTx	Public CTransaction	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_METHODS int Public		<i>Default:</i>
CURRENT_VERSION int32_t Public Static Const		<i>Default:</i> 1

Attribute	Notes	Constraints and tags
hash uint256 Private Const	Memory only.	<i>Default:</i>
nLockTime uint32_t Public Const		<i>Default:</i>
nVersion int32_t Public Const	The local variables are made const to prevent unintended modification without updating the cached hash value. However, CTransaction is not actually immutable; deserialization and assignment are implemented, and bypass the constness. This is safe, as they update the entire structure, including the hash.	<i>Default:</i>
vin std::vector<CTxIn> Public Const		<i>Default:</i>
vout std::vector<CTxOut> Public Const		<i>Default:</i>

Operations

Method	Notes	Parameters
CalculateModifiedSize() unsigned int Public		unsigned int [in] nTxSize
ComputePriority() double Public		double [in] dPriorityInputs unsigned int [in] nTxSize
CTransaction()		

Method	Notes	Parameters
Public		
CTransaction() Public		CMutableTransaction& [in] tx
Const GetHash() uint256& Public		
GetValueOut() CAmount Public		
IsCoinBase() bool Public		
IsNull() bool Public		
operator!=() bool Public		CTransaction& [in] a CTransaction& [in] b
operator=() CTransaction& Public		CTransaction& [in] tx
operator==() bool Public		CTransaction& [in] a CTransaction& [in] b
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
ToString() std::string Public		
UpdateHash() void Private		

CTransactionSignatureSerializer

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {B8397355-4788-44cf-91F6-DFF90BD6D81A}

Wrapper that serializes like CTransaction, but with the modifications required for the signature hash done in-place

Custom Properties

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTransactionSignature Serializer	Private scriptCode CScript	
Association Source -> Destination	Public CTransactionSignature Serializer	Private txTo CTransaction	

Attributes

Attribute	Notes	Constraints and tags
fAnyoneCanPay bool Private Const	! whether the hashtype has the SIGHASH_ANYONECANPAY flag set	<i>Default:</i>
fHashNone bool Private Const	! whether the hashtype is SIGHASH_NONE	<i>Default:</i>
fHashSingle bool Private Const	! whether the hashtype is SIGHASH_SINGLE	<i>Default:</i>
nIn unsigned int Private Const	! input index of txTo being signed	<i>Default:</i>

Attribute	Notes	Constraints and tags
scriptCode CScript& Private Const	! output script being consumed	<i>Default:</i>
txTo CTransaction& Private Const	! reference to the spending transaction (the one being serialized)	<i>Default:</i>

Operations

Method	Notes	Parameters
CTransactionSignatureSerializer() Public		CTransaction& [in] txToIn CScript& [in] scriptCodeIn unsigned int [in] nInIn int [in] nHashTypeIn
Serialize() void Public		S& [in] s int [in] nType int [in] nVersion
SerializeInput() void Public		S& [in] s unsigned int [in] nInput int [in] nType int [in] nVersion
SerializeOutput() void Public		S& [in] s unsigned int [in] nOutput int [in] nType int [in] nVersion

Method	Notes	Parameters
SerializeScriptCode() void Public		S& [in] s int [in] nType int [in] nVersion

CTranslationInterface

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {7F2270FA-2AD5-43df-A2C1-BAF02DFCDCFF}

Signals for translation.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
Translate boost::signals2::signal<std::string (const char* psz)> Public	Translate a message to the native language of the user.	<i>Default:</i>

CTxDestination

Type: Class boost::variant<CNoDestination, CKeyID, CScriptID>
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D578C770-B077-4b0e-905C-4F8172B16CD4}

A txout script template with a specific destination. It is either: CNoDestination: no destination set CKeyID: TX_PUBKEYHASH destination CScriptID: TX_SCRIPTHASH destination A CTxDestination is the internal data type encoded in a CBitcoinAddress

Custom Properties

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public COutputEntry	Public destination CTxDestination	
Association Source -> Destination	Public CCoinControl	Public destChange CTxDestination	

CTxIn**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {16F4B507-5ABC-4b72-B54D-AF7D6097AB8B}

An input of a transaction. It contains the location of the previous transaction's output that it claims and a signature that matches the output's public key.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTxIn	Public nSequence uint32_t	
Association Source -> Destination	Public CTxIn	Public prevout COutPoint	
Association Source -> Destination	Public CTxIn	Public scriptSig CScript	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
nSequence uint32_t Public		<i>Default:</i>
prevout COutPoint Public		<i>Default:</i>
scriptSig CScript Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CTxIn() Public		
CTxIn() Public		COutPoint [in] prevoutIn CScript [in] scriptSigIn uint32_t [in] nSequenceIn
CTxIn() Public		uint256 [in] hashPrevTx uint32_t [in] nOut CScript [in] scriptSigIn uint32_t [in] nSequenceIn

Method	Notes	Parameters
IsFinal() bool Public		
operator!=() bool Public		CTxIn& [in] a CTxIn& [in] b
operator==() bool Public		CTxIn& [in] a CTxIn& [in] b
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
ToString() std::string Public		

CTxInUndo

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {0F9CC87E-FD57-4d7a-8D49-AC2F4132BD68}

Undo information for a CTxIn. Contains the prevout's CTxOut being spent, and if this was the last output of the affected transaction, its metadata as well (coinbase or not, height, transaction version)

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTxInUndo	Public txout CTxOut	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
fCoinBase bool Public	if the outpoint was the last unspent: whether it belonged to a coinbase	<i>Default:</i>
nHeight unsigned int Public	if the outpoint was the last unspent: its height	<i>Default:</i>
nVersion int Public	if the outpoint was the last unspent: its version	<i>Default:</i>
txout CTxOut Public	the txout data before being spent	<i>Default:</i>

Operations

Method	Notes	Parameters
CTxInUndo() Public		
CTxInUndo() Public		CTxOut& [in] txoutIn bool [in] fCoinBaseIn unsigned int [in] nHeightIn int [in] nVersionIn
GetSerializeSize() unsigned int Public		int [in] nType int [in] nVersion
Serialize() void		Stream& [in] s

Method	Notes	Parameters
Public		int [in] nType int [in] nVersion
Unserialize() void Public		Stream& [in] s int [in] nType int [in] nVersion

CTxMemPool

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {DBCEAA17-2F6E-4aee-8ED1-B25D08635D43}

CTxMemPool stores valid-according-to-the-current-best-chain transactions that may be included in the next block. Transactions are added when they are seen on the network (or created by the local node), but not all transactions seen are added to the pool: if a new transaction double-spends an input of a transaction in the pool, it is dropped, as are non-standard transactions.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTxMemPool	Private cachedInnerUsage uint64_t	
Association Source -> Destination	Public CTxMemPool	Private minerPolicyEstimator CBlockPolicyEstimator	
Association Source -> Destination	Public CTxMemPool	Private totalTxSize uint64_t	
Association Source -> Destination	Public CCoinsViewMemPool	Protected mempool CTxMemPool	

Attributes

Attribute	Notes	Constraints and tags
cachedInnerUsage uint64_t Private	! sum of dynamic memory usage of all the map elements (NOT the maps themselves)	<i>Default:</i>
cs CCriticalSection Public		<i>Default:</i> [<u>mutable</u> = true]
fSanityCheck bool Private	! Normally false, true if -checkmempool or -regtest	<i>Default:</i>
mapDeltas std::map<uint256, std::pair<double, CAmount> > Public		<i>Default:</i>
mapNextTx std::map<COutPoint, CInPoint> Public		<i>Default:</i>
mapTx std::map<uint256, CTxMemPoolEntry> Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
minerPolicyEstimator CBlockPolicyEstimator* Private		<i>Default:</i>
nTransactionsUpdated unsigned int Private		<i>Default:</i>
totalTxSize uint64_t Private	! sum of all mempool tx' byte sizes	<i>Default:</i>

Operations

Method	Notes	Parameters
AddTransactionsUpdated() void Public		unsigned int [in] n
addUnchecked() bool Public		uint256& [in] hash CTxMemPoolEntry& [in] entry bool [in] fCurrentEstimate
ApplyDeltas() void Public		uint256 [in] hash double& [in] dPriorityDelta CAmount& [in] nFeeDelta
check() void Public		CCoinsViewCache* [in] pcoins
clear() void Public		
ClearPrioritisation() void Public		uint256 [in] hash

Method	Notes	Parameters
CTxMemPool() Public		CFeeRate& [in] _minRelayFee
~CTxMemPool() Public		
DynamicMemoryUsage() size_t Public		
estimateFee() CFeeRate Public		int [in] nBlocks
estimatePriority() double Public		int [in] nBlocks
exists() bool Public		uint256 [in] hash
GetTotalTxSize() uint64_t Public		
GetTransactionsUpdated () unsigned int Public		
HasNoInputsOf() bool Public		CTransaction& [in] tx
lookup() bool Public		uint256 [in] hash CTransaction& [in] result
PrioritiseTransaction() void Public		uint256 [in] hash std::string [in] strHash double [in] dPriorityDelta CAmount& [in] nFeeDelta
pruneSpent() void Public		uint256& [in] hash CCoins& [in] coins
queryHashes() void Public		std::vector<uint256>& [in] vtxid
ReadFeeEstimates() bool Public		CAutoFile& [in] filein
remove() void Public		CTransaction& [in] tx std::list<CTransaction>& [in] removed

Method	Notes	Parameters
		bool [in] fRecursive
removeCoinbaseSpends() void Public		CCoinsViewCache* [in] pcoins unsigned int [in] nMemPoolHeight
removeConflicts() void Public		CTransaction& [in] tx std::list<CTransaction>& [in] removed
removeForBlock() void Public	Called when a block is connected. Removes from mempool and updates the miner fee estimator.	std::vector<CTransaction>& [in] vtx unsigned int [in] nBlockHeight std::list<CTransaction>& [in] conflicts bool [in] fCurrentEstimate
setSanityCheck() void Public		bool [in] _fSanityCheck
size() unsigned long Public		
WriteFeeEstimates() bool Public		CLogFile& [in] fileout

CTxMemPoolEntry

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {A710B1D1-F6E7-492e-BB7F-F134E566FFB8}

CTxMemPool stores these:

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTxMemPoolEntry	Private nTime int64_t	
Association Source -> Destination	Public CTxMemPoolEntry	Private nFee CAmount	
Association Source -> Destination	Public CTxMemPoolEntry	Private tx CTransaction	

Attributes

Attribute	Notes	Constraints and tags
dPriority double Private	! Priority when entering the mempool	<i>Default:</i>
hadNoDependencies bool Private	! Not dependent on any other txs when it entered the mempool	<i>Default:</i>
nFee CAmount Private	! Cached to avoid expensive parent-transaction lookups	<i>Default:</i>
nHeight unsigned int Private	! Chain height when entering the mempool	<i>Default:</i>
nModSize size_t Private	! ... and modified size for priority	<i>Default:</i>

Attribute	Notes	Constraints and tags
nTime int64_t Private	! Local time when entering the mempool	<i>Default:</i>
nTxSize size_t Private	! ... and avoid recomputing tx size	<i>Default:</i>
nUsageSize size_t Private	! ... and total memory usage	<i>Default:</i>
tx CTransaction Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CTxMemPoolEntry() Public		CTransaction& [in] _tx CAmount& [in] _nFee int64_t [in] _nTime double [in] _dPriority unsigned int [in] _nHeight bool [in] poolHasNoInputsOf
CTxMemPoolEntry() Public		
CTxMemPoolEntry() Public		CTxMemPoolEntry& [in] other

Method	Notes	Parameters
DynamicMemoryUsage() size_t Public		
GetFee() CAmount Public		
GetHeight() unsigned int Public		
GetPriority() double Public		unsigned int [in] currentHeight
GetTime() int64_t Public		
Const GetTx() CTransaction& Public		
GetTxSize() size_t Public		
WasClearAtEntry() bool Public		

CTxOut

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {406F340B-CEE0-4f08-B2E8-44840DA793F5}

An output of a transaction. It contains the public key that the next input must be able to sign with to claim it.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTxOut	Public scriptPubKey CScript	
Association Source -> Destination	Public CTxOut	Public nValue CAmount	
Association Source -> Destination	Public CTxOutCompressor	Private txout CTxOut	
Association Source -> Destination	Public CTxInUndo	Public txout CTxOut	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_METHODS int Public		<i>Default:</i>
nValue CAmount Public		<i>Default:</i>
scriptPubKey CScript Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CTxOut() Public		
CTxOut() Public		CAmount& [in] nValueIn CScript [in] scriptPubKeyIn
GetDustThreshold() CAmount Public		CFeeRate& [in] minRelayTxFee
GetHash() uint256 Public		
IsDust() bool Public		CFeeRate& [in] minRelayTxFee
IsNull() bool Public		
operator!=() bool Public		CTxOut& [in] a CTxOut& [in] b
operator==() bool		CTxOut& [in] a

Method	Notes	Parameters
Public		CTxOut& [in] b
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		
ToString() std::string Public		

CTxOutCompressor

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {85DD77A2-8850-4ab8-A3F9-2F3F589B3D4C}

wrapper for CTxOut that provides a more compact serialization

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTxOutCompressor	Private txout CTxOut	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
txout CTxOut& Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Static CompressAmount() uint64_t Public	Amount compression: If the amount is 0, output 0 first, divide the amount (in base units) by the largest power of 10 possible; call the exponent e (e is max 9) if e<9, the last digit of the resulting number cannot be 0; store it as d, and drop it (divide by 10) call the result n output $1 + 10^{*(9*n + d - 1)} + e$ if e==9, we only know the resulting number is not zero, so output $1 + 10^{*(n - 1)} + 9$ (this is decodable, as d is in [1-9] and e is in [0-9])	uint64_t [in] nAmount
CTxOutCompressor() Public		CTxOut& [in] txoutIn
Static DecompressAmount() uint64_t Public		uint64_t [in] nAmount
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion

CTxUndo**Type:****Class****Status:**

Proposed. Version 1.0. Phase 1.0.

Package:Class Model **Keywords:****Detail:**

Created on 8/30/2015. Last modified on 8/30/2015.

GUID:

{8EC771BC-857B-4eb7-9A77-4B40A2218BFE}

Undo information for a CTransaction

Custom Properties

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
vprevout std::vector<CTxInUndo> Public	undo information for all txins	<i>Default:</i>

Operations

Method	Notes	Parameters
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion

CUnsignedAlert

Type:	Class
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model <i>Keywords:</i>
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>
GUID:	{436A7C1E-EDBE-43a0-A334-F77538B2DC76}

Alerts are for notifying old versions if they become too obsolete and need to upgrade. The message is displayed in the status bar. Alert messages are broadcast as a vector of signed data. Unserializing may not read the entire buffer if the alert is for a newer version, but older versions can still relay the original data.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CUnsignedAlert	Public nExpiration int64_t	
Association Source -> Destination	Public CUnsignedAlert	Public nRelayUntil int64_t	
Generalization Source -> Destination	Public CArt	Public CUnsignedAlert	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
nCancel int Public		<i>Default:</i>
nExpiration int64_t Public		<i>Default:</i>
nID int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nMaxVer int Public	highest version inclusive	<i>Default:</i>
nMinVer int Public	lowest version inclusive	<i>Default:</i>
nPriority int Public		<i>Default:</i>
nRelayUntil int64_t Public	when newer nodes stop relaying to newer nodes	<i>Default:</i>
nVersion int Public		<i>Default:</i>
setCancel std::set<int> Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
setSubVer std::set<std::string> Public	empty matches all	<i>Default:</i>
strComment std::string Public	Actions	<i>Default:</i>
strReserved std::string Public		<i>Default:</i>
strStatusBar std::string Public		<i>Default:</i>

Operations

Method	Notes	Parameters
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
SetNull() void Public		
ToString() std::string Public		

CValidationInterface

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {50D9F599-06AF-47bc-8079-1FB9B79A3EEC}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CWallet	Public CValidationInterface	

Operations

Method	Notes	Parameters
BlockChecked() void Protected		CBlock& [in] prm1 CValidationState& [in] prm2
GetScriptForMining() void Protected		boost::shared_ptr<CReserveScript>& [in] prm1
Inventory() void Protected		uint256& [in] hash
RegisterValidationInterface() Protected		CValidationInterface* [in] prm1
ResendWalletTransactions() void Protected		int64_t [in] nBestBlockTime
ResetRequestCount() void Protected		uint256& [in] hash
SetBestChain() void Protected		CBlockLocator& [in] locator
SyncTransaction() void Protected		CTransaction& [in] tx CBlock* [in] pblock
UnregisterAllValidationInterfaces() Protected		

Method	Notes	Parameters
UnregisterValidationInterface() Protected		CValidationInterface* [in] prm1
UpdatedTransaction() void Protected		uint256& [in] hash

CValidationState

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model Keywords:

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {0191A2D6-6270-445f-9F69-A7541BD2AAFE}

Capture information about block/transaction validation

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
chRejectCode unsigned int Private		<i>Default:</i>
corruptionPossible bool Private		<i>Default:</i>
nDoS int Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
strDebugMessage std::string Private		<i>Default:</i>
strRejectReason std::string Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CorruptionPossible() bool Public		
CValidationState() Public		
DoS() bool Public		int [in] level bool [in] ret unsigned int [in] chRejectCodeIn std::string& [in] strRejectReasonIn bool [in] corruptionIn std::string& [in] strDebugMessageIn
Error() bool Public		std::string& [in] strRejectReasonIn
GetDebugMessage() std::string Public		
GetRejectCode() unsigned int Public		
GetRejectReason() std::string Public		
Invalid() bool Public		bool [in] ret unsigned int [in] _chRejectCode

Method	Notes	Parameters
		std::string& [in] _strRejectReason std::string& [in] _strDebugMessage
IsError() bool Public		
IsValid() bool Public		
IsValid() bool Public		int& [in] nDoSOut
IsValid() bool Public		

mode_state

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {6D93F550-1FEE-4096-8AD2-F16EA15A5D57}

Custom Properties

- isActive = False

Tagged Values

- typeSynonyms = mode.

Attributes

Attribute	Notes	Constraints and tags
MODE_VALID Private	! everything ok	<i>Default:</i>

Attribute	Notes	Constraints and tags
MODE_INVALID Private	! network rule violation (DoS value may be set)	<i>Default:</i>
MODE_ERROR Private	! run-time error	<i>Default:</i>

CVarInt

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {FCC16601-88E0-46a5-AC80-3A9080DD4F4D}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
n I& Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
CVarInt() Public		I& [in] nIn

Method	Notes	Parameters
GetSerializeSize() unsigned int Public		int [in] prm1 int [in] prm2
Serialize() void Public		Stream& [in] s int [in] prm1 int [in] prm2
Unserialize() void Public		Stream& [in] s int [in] prm1 int [in] prm2

CVerifyDB

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {C4D27501-F9B8-4db2-92DF-69B4D3DEB581}

RAII wrapper for VerifyDB: Verify consistency of the block and coin databases

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
CVerifyDB() Public		
~CVerifyDB() Public		
VerifyDB() bool Public		CCoinsView* [in] coinsview int [in] nCheckLevel int [in] nCheckDepth

CWallet

Type:	Class	CCryptoKeyStore, CValidationInterface
Status:	Proposed.	Version 1.0. Phase 1.0.
Package:	Class Model	<i>Keywords:</i>
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>	
GUID:	{212C9A3C-8BAF-4a96-9493-BE6444262306}	

A CWallet is an extension of a keystore, which also maintains a set of transactions and balances, and provides the ability to create new transactions.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CReserveKey	Protected pwallet CWallet	
Generalization Source -> Destination	Public CWallet	Public CValidationInterface	
Generalization Source -> Destination	Public CWallet	Public CCryptoKeyStore	
Association Source -> Destination	Public CWallet	Public minTxFee CFeeRate	
Association Source -> Destination	Public CWallet	Private nLastResend int64_t	
Association Source -> Destination	Public CWallet	Private nNextResend int64_t	
Association Source -> Destination	Public CWallet	Public nOrderPosNext int64_t	
Association Source -> Destination	Public CWallet	Public nTimeFirstKey int64_t	
Association Source -> Destination	Public CWallet	Private pwalletdbEncryption CWalletDB	
Association Source -> Destination	Public CWallet	Public vchDefaultKey CPubKey	
Association Source -> Destination	Public CWalletTx	Private pwallet CWallet	
Association Source -> Destination	Public CWallet	Public mapMasterKeys MasterKeyMap	

Connector	Source	Target	Notes
Association Source -> Destination	Public CWallet	Private mapTxSpends TxSpends	

Attributes

Attribute	Notes	Constraints and tags
cs_wallet CCriticalSection Public	Main wallet lock. This lock protects all the fields added by CWallet except for: fFileBacked (immutable after instantiation) strWalletFile (immutable after instantiation)	<i>Default:</i> [<u>mutable</u> = true]
fBroadcastTransactions bool Private		<i>Default:</i>
fFileBacked bool Public		<i>Default:</i>
mapAddressBook std::map<CTxDestination, CAddressBookData> Public		<i>Default:</i>
mapKeyMetadata std::map<CKeKeyID, CKeyMetadata> Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
mapMasterKeys MasterKeyMap Public		<i>Default:</i>
mapRequestCount std::map<uint256, int> Public		<i>Default:</i>
mapTxSpends TxSpends Private		<i>Default:</i>
mapWallet std::map<uint256, CWalletTx> Public		<i>Default:</i>
minTxFee CFeeRate Public Static		<i>Default:</i>
nLastResend int64_t Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
nMasterKeyMaxID unsigned int Public		<i>Default:</i>
nNextResend int64_t Private		<i>Default:</i>
nOrderPosNext int64_t Public		<i>Default:</i>
NotifyAddressBookChanged boost::signals2::signal<void (CWallet *wallet, const CTxDestination &address, const std::string &label, bool isMine, const std::string &purpose, ChangeType status)> Public	Address book entry changed. @note called with lock cs_wallet held.	<i>Default:</i>
NotifyTransactionChanged boost::signals2::signal<void (CWallet *wallet, const uint256 &hashTx, ChangeType status)> Public	Wallet transaction added, removed or updated. @note called with lock cs_wallet held.	<i>Default:</i>

Attribute	Notes	Constraints and tags
NotifyWatchonlyChange d boost::signals2::signal<void (bool fHaveWatchOnly)> Public	Watch-only address added	<i>Default:</i>
nTimeFirstKey int64_t Public		<i>Default:</i>
nWalletMaxVersion int Private	! the maximum wallet format version: memory-only variable that specifies to what version this wallet may be upgraded	<i>Default:</i>
nWalletVersion int Private	! the current wallet version: clients below this version are not able to load the wallet	<i>Default:</i>
pwalletdbEncryption CWalletDB* Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
setKeyPool std::set<int64_t> Public		<i>Default:</i>
setLockedCoins std::set<COutPoint> Public		<i>Default:</i>
ShowProgress boost::signals2::signal<void (const std::string &title, int nProgress)> Public	Show progress e.g. for rescan	<i>Default:</i>
strWalletFile std::string Public		<i>Default:</i>
vchDefaultKey CPubKey Public		<i>Default:</i>

Operations

Method	Notes	Parameters
AddCryptedKey() bool Public		CPubKey& [in] vchPubKey std::vector<unsigned char>& [in] vchCryptdSecret
AddCScript() bool		CScript& [in] redeemScript

Method	Notes	Parameters
Public		
AddDestData() bool Public		CTxDestination& [in] dest std::string& [in] key std::string& [in] value
AddKeyPubKey() bool Public		CKey& [in] key CPubKey& [in] pubkey
AddToSpends() void Private		COutPoint& [in] outpoint uint256& [in] wtxid
AddToSpends() void Private		uint256& [in] wtxid
AddToWallet() bool Public		CWalletTx& [in] wtxIn bool [in] fFromLoadWallet CWalletDB* [in] pwalletdb
AddToWalletIfInvolvingMe() bool Public	Add a transaction to the wallet, or update it. pblock is optional, but should be provided if the transaction is known to be in a block. If fUpdate is true, existing transactions will be updated.	CTransaction& [in] tx CBlock* [in] pblock bool [in] fUpdate
AddWatchOnly() bool Public		CScript& [in] dest
AvailableCoins() void Public		std::vector<COutput>& [in] vCoins bool [in] fOnlyConfirmed CCoinControl* [in] coinControl bool [in] fIncludeZeroValue
CanSupportFeature() bool Public		enum WalletFeature [in] wf
ChangeWalletPassphrase() bool Public		SecureString& [in] strOldWalletPassphrase

Method	Notes	Parameters
		SecureString& [in] strNewWalletPassphrase
CommitTransaction() bool Public	Call after CreateTransaction unless you want to abort	CWalletTx& [in] wtxNew CReserveKey& [in] reservekey
CreateTransaction() bool Public		std::vector<CRecipient>& [in] vecSend CWalletTx& [in] wtxNew CReserveKey& [in] reservekey CAmount& [in] nFeeRet int& [in] nChangePosRet std::string& [in] strFailReason CCoinControl* [in] coinControl bool [in] sign
CWallet() Public		
CWallet() Public		std::string& [in] strWalletFileIn
~CWallet() Public		
DelAddressBook() bool Public		CTxDestination& [in] address
EncryptWallet() bool Public		SecureString& [in] strWalletPassphrase
EraseDestData() bool Public		CTxDestination& [in] dest std::string& [in] key
Flush() void Public		bool [in] shutdown
FundTransaction() bool Public		CMutableTransaction& [in] tx CAmount& [in] nFeeRet int& [in] nChangePosRet

Method	Notes	Parameters
		std::string& [in] strFailReason bool [in] includeWatching
GenerateNewKey() CPubKey Public		
GetAccountAddresses() std::set<CTxDestination> Public		std::string& [in] strAccount
GetAddressBalances() std::map<CTxDestination, CAmount> Public		
GetAddressGroupings() std::set< std::set<CTxDestination> > Public		
GetAllReserveKeys() void Public		std::set<CKeyID>& [in] setAddress
GetBalance() CAmount Public		
GetBroadcastTransactions() bool Public		
GetChange() CAmount Public		CTxOut& [in] txout
GetChange() CAmount Public		CTransaction& [in] tx
GetConflicts() std::set<uint256> Public		uint256& [in] txid
GetCredit() CAmount Public		CTxOut& [in] txout isminefilter& [in] filter
GetCredit() CAmount Public		CTransaction& [in] tx isminefilter& [in] filter
GetDebit() CAmount Public		CTxIn& [in] txin isminefilter& [in] filter
GetDebit() CAmount		CTransaction& [in] tx

Method	Notes	Parameters
Public		isminefilter& [in] filter
GetDestData() bool Public		CTxDestination& [in] dest std::string& [in] key std::string* [in] value
GetImmatureBalance() CAmount Public		
GetImmatureWatchOnlyBalance() CAmount Public		
GetKeyBirthTimes() void Public		std::map<CKeyID, int64_t>& [in] mapKeyBirth
GetKeyFromPool() bool Public		CPubKey& [in] key
GetKeyPoolSize() unsigned int Public		
Static GetMinimumFee() CAmount Public		unsigned int [in] nTxBytes unsigned int [in] nConfirmTarget CTxMemPool& [in] pool
GetOldestKeyPoolTime() int64_t Public		
GetScriptForMining() void Public		boost::shared_ptr<CReserveScript>& [in] script
GetUnconfirmedBalance() CAmount Public		
GetUnconfirmedWatchOnlyBalance() CAmount Public		
GetVersion() int Public		
Const GetWalletTx() CWalletTx* Public		uint256& [in] hash
GetWatchOnlyBalance() CAmount		

Method	Notes	Parameters
Public		
IncOrderPosNext() int64_t Public		CWalletDB* [in] pwalletdb
Inventory() void Public		uint256& [in] hash
IsChange() bool Public		CTxOut& [in] txout
IsFromMe() bool Public		CTransaction& [in] tx
IsLockedCoin() bool Public		uint256 [in] hash unsigned int [in] n
IsMine() isminetype Public		CTxIn& [in] txin
IsMine() isminetype Public		CTxOut& [in] txout
IsMine() bool Public		CTransaction& [in] tx
IsSpent() bool Public	Outpoint is spent if any non-conflicted transaction spends it:	uint256& [in] hash unsigned int [in] n
KeepKey() void Public		int64_t [in] nIndex
ListLockedCoins() void Public		std::vector<COutPoint>& [in] vOutpts
LoadCryptedKey() bool Public		CPubKey& [in] vchPubKey std::vector<unsigned char>& [in] vchCryptedSecret
LoadCScript() bool Public		CScript& [in] redeemScript
LoadDestData() bool Public		CTxDestination& [in] dest std::string& [in] key std::string& [in] value

Method	Notes	Parameters
LoadKey() bool Public		CKey& [in] key CPubKey& [in] pubkey
LoadKeyMetadata() bool Public		CPubKey& [in] pubkey CKeyMetadata& [in] metadata
LoadMinVersion() bool Public		int [in] nVersion
LoadWallet() DBErrors Public		bool& [in] fFirstRunRet
LoadWatchOnly() bool Public		CScript& [in] dest
LockCoin() void Public		COutPoint& [in] output
MarkDirty() void Public		
NewKeyPool() bool Public	Mark old keypool keys as used, and generate all new keys	
OrderedTxItems() TxItems Public		std::list<CAccountingEntry>& [in] acentries std::string [in] strAccount
ReacceptWalletTransactions() void Public		
RemoveWatchOnly() bool Public		CScript& [in] dest
ResendWalletTransactions() void Public		int64_t [in] nBestBlockTime
ResendWalletTransactionsBefore() std::vector<uint256> Public		int64_t [in] nTime
ReserveKeyFromKeyPool() void Public		int64_t& [in] nIndex CKeyPool& [in] keypool
ResetRequestCount() void Public		uint256& [in] hash

Method	Notes	Parameters
ReturnKey() void Public		int64_t [in] nIndex
ScanForWalletTransactions() int Public	Scan the block chain (starting in pindexStart) for transactions from or to us. If fUpdate is true, found transactions that already exist in the wallet will be updated.	CBlockIndex* [in] pindexStart bool [in] fUpdate
SelectCoins() bool Private		CAmount& [in] nTargetValue std::set<std::pair<const CWalletTx*,unsigned int> & [in] setCoinsRet CAmount& [in] nValueRet CCoinControl* [in] coinControl
SelectCoinsMinConf() bool Public		CAmount& [in] nTargetValue int [in] nConfMine int [in] nConfTheirs std::vector<COOutput> [in] vCoins std::set<std::pair<const CWalletTx*,unsigned int> & [in] setCoinsRet CAmount& [in] nValueRet
SetAddressBook() bool Public		CTxDestination& [in] address std::string& [in] strName std::string& [in] purpose
SetBestChain() void Public		CBlockLocator& [in] loc
SetBroadcastTransaction() void Public		bool [in] broadcast
SetDefaultKey() bool Public		CPubKey& [in] vchPubKey
SetMaxVersion() bool Public		int [in] nVersion
SetMinVersion() bool Public		enum WalletFeature [in] nVersion

Method	Notes	Parameters
		CWalletDB* [in] pwalletdbIn bool [in] fExplicit
SetNull() void Public		
SyncMetaData() void Private		std::pair<TxSpends::iterator, TxSpends::iterator> [in] prm1
SyncTransaction() void Public		CTransaction& [in] tx CBlock* [in] pblock
TopUpKeyPool() bool Public		unsigned int [in] kpSize
Unlock() bool Public		SecureString& [in] strWalletPassphrase
UnlockAllCoins() void Public		
UnlockCoin() void Public		COutPoint& [in] output
UpdatedTransaction() void Public		uint256& [in] hashTx
Static Verify() bool Public		std::string& [in] walletFile std::string& [in] warningString std::string& [in] errorString
ZapWalletTx() DBErrors Public		std::vector<CWalletTx>& [in] vWtx

MasterKeyMap

Type: **Class** **std::map<unsigned int, CMasterKey>**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {EADA2B94-BE54-4714-8565-AF34B44BC34D}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CWallet	Public mapMasterKeys MasterKeyMap	

TxItems

Type: **Class** [std::multimap<int64 t, TxPair >](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {5EBB7F99-9697-423d-AED3-BB650397915F}

Custom Properties

- isActive = False

TxPair

Type: **Class** [std::pair<CWalletTx*, CAccountingEntry*>](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {E9B52F0B-B4B6-4357-9AAD-1708D2DDBE00}

Custom Properties

- isActive = False

TxSpends

Type: **Class** [std::multimap<COutPoint, uint256>](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {F912E6F4-3CAC-4d23-8F04-52CF0BDF689E}

Used to keep track of spent outpoints, and detect and report conflicts (double-spends or mutated transactions where the mutant gets mined).

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CWallet	Private mapTxSpends TxSpends	

CWalletDB

Type: **Class** **CDB**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {7DB4DE89-E227-403f-A499-FF8ED7770030}

Access to the wallet database (wallet.dat)

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CWalletDB	Public CDB	
Association Source -> Destination	Public CWallet	Private pwalletdbEncryption CWalletDB	

Operations

Method	Notes	Parameters
CWalletDB() Public		std::string& [in] strFilename char* [in] pszMode bool [in] fFlushOnClose

Method	Notes	Parameters
CWalletDB() Private		CWalletDB& [in] prm1
EraseDestData() bool Public		std::string& [in] address std::string& [in] key
EraseName() bool Public		std::string& [in] strAddress
ErasePool() bool Public		int64_t [in] nPool
ErasePurpose() bool Public		std::string& [in] strAddress
EraseTx() bool Public		uint256 [in] hash
EraseWatchOnly() bool Public		CScript& [in] script
FindWalletTx() DBErrors Public		CWallet* [in] pwallet std::vector<uint256>& [in] vTxHash std::vector<CWalletTx>& [in] vWtx
GetAccountCreditDebit() CAmount Public		std::string& [in] strAccount
ListAccountCreditDebit() void Public		std::string& [in] strAccount std::list<CAccountingEntry>& [in] acentries
LoadWallet() DBErrors Public		CWallet* [in] pwallet
operator=() void Private		CWalletDB& [in] prm1
ReadAccount() bool Public		std::string& [in] strAccount CAccount& [in] account
ReadBestBlock() bool Public		CBlockLocator& [in] locator

Method	Notes	Parameters
ReadPool() bool Public		int64_t [in] nPool CKeyPool& [in] keypool
Static Recover() bool Public	Try to (very carefully!) recover wallet.dat if there is a problem.	CDBEnv& [in] dbenv std::string& [in] filename bool [in] fOnlyKeys
Static Recover() bool Public		CDBEnv& [in] dbenv std::string& [in] filename
ReorderTransactions() DBErrors Public		CWallet* [in] pwallet
WriteAccount() bool Public		std::string& [in] strAccount CAccount& [in] account
WriteAccountingEntry() bool Public		CAccountingEntry& [in] acentry
WriteAccountingEntry() bool Private		uint64_t [in] nAccEntryNum CAccountingEntry& [in] acentry
WriteBestBlock() bool Public		CBlockLocator& [in] locator
WriteCryptedKey() bool Public		CPubKey& [in] vchPubKey std::vector<unsigned char>& [in] vchCryptedSecret CKeyMetadata& [in] keyMeta
WriteCScript() bool Public		uint160& [in] hash CScript& [in] redeemScript
WriteDefaultKey() bool Public		CPubKey& [in] vchPubKey
WriteDestData() bool Public		std::string& [in] address std::string& [in] key

Method	Notes	Parameters
		std::string& [in] value
WriteKey() bool Public		CPubKey& [in] vchPubKey CPrivateKey& [in] vchPrivKey CKeyMetadata& [in] keyMeta
WriteMasterKey() bool Public		unsigned int [in] nID CMasterKey& [in] kMasterKey
WriteMinVersion() bool Public		int [in] nVersion
WriteName() bool Public		std::string& [in] strAddress std::string& [in] strName
WriteOrderPosNext() bool Public		int64_t [in] nOrderPosNext
WritePool() bool Public		int64_t [in] nPool CKeyPool& [in] keypool
WritePurpose() bool Public		std::string& [in] strAddress std::string& [in] purpose
WriteTx() bool Public		uint256 [in] hash CWalletTx& [in] wtx
WriteWatchOnly() bool Public		CScript& [in] script
ZapWalletTx() DBErrors Public		CWallet* [in] pwallet std::vector<CWalletTx>& [in] vWtx

CWalletKey

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {3A7632BE-0ACE-45c1-B9CD-7DCFD62C5A61}

Private key that includes an expiration date in case it never gets used.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CWalletKey	Public nTimeCreated int64_t	
Association Source -> Destination	Public CWalletKey	Public nTimeExpires int64_t	
Association Source -> Destination	Public CWalletKey	Public vchPrivKey CPrivKey	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
nTimeCreated int64_t Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nTimeExpires int64_t Public		<i>Default:</i>
strComment std::string Public		<i>Default:</i>
vchPrivKey CPrivateKey Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CWalletKey() Public		int64_t [in] nExpires
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion

CWalletScanState

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D7CAC45F-431E-470f-81D3-E56166E79AB4}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
fAnyUnordered bool Public		<i>Default:</i>
fIsEncrypted bool Public		<i>Default:</i>
nCKeys unsigned int Public		<i>Default:</i>
nFileVersion int Public		<i>Default:</i>
nKeyMeta unsigned int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nKeys unsigned int Public		<i>Default:</i>
vWalletUpgrade vector<uint256> Public		<i>Default:</i>

Operations

Method	Notes	Parameters
CWalletScanState() Public		

CWalletTx

Type: [Class](#) [CMerkleTx](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {B12F9A99-0B11-4b3a-9CDB-00BA09D715DA}

A transaction with a bunch of additional info that only the owner cares about. It includes any unrecorded transactions needed to link it back to the block chain.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public WalletModelTransactio n	Private walletTransaction CWalletTx	
Association Source -> Destination	Public CWalletTx	Public nChangeCached CAmount	
Association Source -> Destination	Public CWalletTx	Public mapValue mapValue_t	

Connector	Source	Target	Notes
Association Source -> Destination	Public CWalletTx	Public nAvailableWatchCredit Cached CAmount	
Association Source -> Destination	Public CWalletTx	Public nOrderPos int64_t	
Association Source -> Destination	Public CWalletTx	Public nCreditCached CAmount	
Association Source -> Destination	Public CWalletTx	Public nDebitCached CAmount	
Association Source -> Destination	Public CWalletTx	Public nImmatureCreditCache d CAmount	
Association Source -> Destination	Public CWalletTx	Public nImmatureWatchCredit Cached CAmount	
Association Source -> Destination	Public CWalletTx	Public nAvailableCreditCache d CAmount	
Association Source -> Destination	Public CWalletTx	Public nWatchCreditCached CAmount	
Association Source -> Destination	Public CWalletTx	Public nWatchDebitCached CAmount	
Association Source -> Destination	Public CWalletTx	Private pwallet CWallet	
Generalization Source -> Destination	Public CWalletTx	Public CMerkleTx	
Association Source -> Destination	Public COutput	Public tx CWalletTx	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_MET HODS int Public		<i>Default:</i>
fAvailableCreditCached bool Public		<i>Default:</i> [<u>mutable</u> = true]
fAvailableWatchCreditCached bool Public		<i>Default:</i> [<u>mutable</u> = true]
fChangeCached bool Public		<i>Default:</i> [<u>mutable</u> = true]
fCreditCached bool Public		<i>Default:</i> [<u>mutable</u> = true]
fDebitCached bool Public	memory only	<i>Default:</i> [<u>mutable</u> = true]

Attribute	Notes	Constraints and tags
fFromMe char Public		<i>Default:</i>
fImmatureCreditCached bool Public		<i>Default:</i> [<u>mutable</u> = true]
fImmatureWatchCreditCached bool Public		<i>Default:</i> [<u>mutable</u> = true]
fTimeReceivedIsTxTime unsigned int Public		<i>Default:</i>
fWatchCreditCached bool Public		<i>Default:</i> [<u>mutable</u> = true]
fWatchDebitCached bool Public		<i>Default:</i> [<u>mutable</u> = true]

Attribute	Notes	Constraints and tags
mapValue mapValue_t Public		<i>Default:</i>
nAvailableCreditCached CAmount Public		<i>Default:</i> [<u>mutable</u> = true]
nAvailableWatchCreditCached CAmount Public		<i>Default:</i> [<u>mutable</u> = true]
nChangeCached CAmount Public		<i>Default:</i> [<u>mutable</u> = true]
nCreditCached CAmount Public		<i>Default:</i> [<u>mutable</u> = true]
nDebitCached CAmount Public		<i>Default:</i> [<u>mutable</u> = true]

Attribute	Notes	Constraints and tags
nImmatureCreditCached CAmount Public		<i>Default:</i> [<u>mutable</u> = true]
nImmatureWatchCreditCached CAmount Public		<i>Default:</i> [<u>mutable</u> = true]
nOrderPos int64_t Public	! position in ordered transaction list	<i>Default:</i>
nTimeReceived unsigned int Public	! time received by this node	<i>Default:</i>
nTimeSmart unsigned int Public		<i>Default:</i>
nWatchCreditCached CAmount Public		<i>Default:</i> [<u>mutable</u> = true]

Attribute	Notes	Constraints and tags
nWatchDebitCached CAmount Public		<i>Default:</i> [mutable = true]
pwallet CWallet* Private Const		<i>Default:</i>
strFromAccount std::string Public		<i>Default:</i>
vOrderForm std::vector<std::pair<std::string, std::string>> Public		<i>Default:</i>

Operations

Method	Notes	Parameters
BindWallet() void Public		CWallet* [in] pwalletIn
CWalletTx() Public		
CWalletTx() Public		CWallet* [in] pwalletIn
CWalletTx() Public		CWallet* [in] pwalletIn CMerkleTx& [in] txIn
CWalletTx() Public		CWallet* [in] pwalletIn CTransaction& [in] txIn

Method	Notes	Parameters
GetAccountAmounts() void Public		std::string& [in] strAccount CAmount& [in] nReceived CAmount& [in] nSent CAmount& [in] nFee isminefilter& [in] filter
GetAmounts() void Public		std::list<COutputEntry>& [in] listReceived std::list<COutputEntry>& [in] listSent CAmount& [in] nFee std::string& [in] strSentAccount isminefilter& [in] filter
GetAvailableCredit() CAmount Public		bool [in] fUseCache
GetAvailableWatchOnly Credit() CAmount Public		bool& [in] fUseCache
GetChange() CAmount Public		
GetConflicts() std::set<uint256> Public		
GetCredit() CAmount Public		isminefilter& [in] filter
GetDebit() CAmount Public		isminefilter& [in] filter
GetImmatureCredit() CAmount Public		bool [in] fUseCache
GetImmatureWatchOnly Credit() CAmount Public		bool& [in] fUseCache
GetRequestCount() int Public		
GetTxTime() int64_t Public		
Init() void Public		CWallet* [in] pwalletIn

Method	Notes	Parameters
IsEquivalentTo() bool Public		CWalletTx& [in] tx
IsFromMe() bool Public		isminefilter& [in] filter
IsTrusted() bool Public		
MarkDirty() void Public		
RelayWalletTransaction() bool Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion
WriteToDisk() bool Public		CWalletDB* [in] pwalletdb

CachingTransactionSignatureChecker

Type: [Class](#) [TransactionSignatureChecker](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords](#):
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {DEAADE91-6D60-4c67-BABC-40808D5BC329}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CachingTransactionSignatureChecker	Public TransactionSignatureChecker	

Attributes

Attribute	Notes	Constraints and tags
store bool Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CachingTransactionSignatureChecker() Public		CTransaction* [in] txToIn unsigned int [in] nInIn bool [in] storeIn
VerifySignature() bool Public		std::vector<unsigned char>& [in] vchSig CPubKey& [in] vchPubKey uint256& [in] sighash

ChainCode

Type: **Class** **uint256**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model *Keywords:*
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 GUID: {DFE5DEC9-C1BA-41c0-AA2D-78078F991FC7}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public ChainCode	Public uint256	

ChainCode

Type:	Class	uint256
Status:	Proposed.	Version 1.0. Phase 1.0.
Package:	Class Model	<i>Keywords:</i>
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>	
GUID:	{1BD619D4-F4F8-4591-A82A-8F9B11B5C6AD}	

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CExtKey	Public chaincode ChainCode	
Association Source -> Destination	Public CExtPubKey	Public chaincode ChainCode	
Generalization Source -> Destination	Public ChainCode	Public uint256	

ChangeType

Type:	Enumeration
Status:	Proposed.
Package:	Version 1.0. Phase 1.0.
Detail:	<i>Class Model Keywords:</i>
GUID:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>

General change type (added, updated, removed).

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
CT_NEW Public		<i>Default:</i>
CT_UPDATED Public		<i>Default:</i>
CT_DELETED Public		<i>Default:</i>

CoinControlTreeWidget

Type: **Class** [QTreeWidget](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model [Keywords:](#)

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {43293D11-670A-4ee2-9DCA-6C84C375A8FF}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
CoinControlTreeWidget() Public		QWidget* [in] parent
keyPressEvent() void Protected		QKeyEvent* [in] event

CombinerAll

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A8CBC585-6E96-4b77-A596-CDF40C96F9AA}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
operator() bool Public		I [in] first I [in] last

result_type

Type: [Class](#) [bool](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {BBFB934E-17C5-415d-BE81-F0CBA155F55A}

Custom Properties

- isActive = False

CompareValueOnly

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {467510D7-8E60-408c-A148-635BA584B929}

```
@defgroup mapWallet
@{
```

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
operator() bool Public		pair<CAmount, pair<const CWalletTx*, unsigned int>>& [in] t1 pair<CAmount, pair<const CWalletTx*, unsigned int>>& [in] t2

CryptedKeyMap

Type: **Class** [std::map<CKeyID, std::pair<CPubKey, std::vector<unsigned char>>>](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D63F9583-904A-4c92-99FC-3F3A1FA06960}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCryptoKeyStore	Private mapCryptedKeys CryptedKeyMap	

DBErrors

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {0A57010F-ABCC-4a15-A278-BB6EDEE6DFC1}

Error statuses for the wallet database

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
DB_LOAD_OK Public		<i>Default:</i>
DB_CORRUPT Public		<i>Default:</i>
DB_NONCRITICAL_ER_ROR Public		<i>Default:</i>
DB_TOO_NEW Public		<i>Default:</i>
DB_LOAD_FAIL Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
DB_NEED_REWRITE Public		<i>Default:</i>

DummySignatureChecker

Type: Class **BaseSignatureChecker**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {3D6DC47A-6C18-43fb-9750-F9A6F6B17425}

Dummy signature checker which accepts all signatures.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public DummySignatureChecker	Public BaseSignatureChecker	

Operations

Method	Notes	Parameters
CheckSig() bool Public		<code>std::vector<unsigned char>& [in] scriptSig</code> <code>std::vector<unsigned char>& [in] vchPubKey</code> <code>CScript& [in] scriptCode</code>
DummySignatureChecker() Public		

DummySignatureCreator

Type: Class **BaseSignatureCreator**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {B5A8896E-731D-406b-9C9B-88174BE5C6CA}

A signature creator that just produces 72-byte empty signatures.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
<u>Generalization</u> Source -> Destination	Public DummySignatureCreator	Public BaseSignatureCreator	

Operations

Method	Notes	Parameters
Const Checker() BaseSignatureChecker& Public		
CreateSig() bool Public		std::vector<unsigned char>& [in] vchSig CKeyID& [in] keyid CScript& [in] scriptCode
DummySignatureCreator 0 Public		CKeyStore* [in] keystoreIn

HTTPStatusCode

Type: Enumeration
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {CF0001AC-0E5C-4312-A65B-D6516C68C4D1}

! HTTP status codes

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
HTTP_OK Public		<i>Default:</i> 200
HTTP_BAD_REQUEST Public		<i>Default:</i> 400
HTTP_UNAUTHORIZED D Public		<i>Default:</i> 401
HTTP_FORBIDDEN Public		<i>Default:</i> 403
HTTP_NOT_FOUND Public		<i>Default:</i> 404
HTTP_INTERNAL_SERVER_ERROR Public		<i>Default:</i> 500

Attribute	Notes	Constraints and tags
HTTP_SERVICE_UNAVAILABLE Public		<i>Default:</i> 503

HelpMessageMode

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {9EDBF323-EBCB-4414-95FD-7EC265362776}

The help message mode determines what help message to show

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
HMM_BITCOIND Public		<i>Default:</i>
HMM_BITCOIN_QT Public		<i>Default:</i>

KeyMap

Type: [Class std::map<CKeyID, CKey>](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**

Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {9C2EC1BB-354C-43b4-8C75-71743C2F585B}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBasicKeyStore	Protected mapKeys KeyMap	

LimitedString

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {C2B10863-E440-4625-BB92-E7F6DD3E67BE}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
string std::string& Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
GetSerializeSize() unsigned int Public		int [in] prm1 int [in] prm2

Method	Notes	Parameters
LimitedString() Public		std::string& [in] string
Serialize() void Public		Stream& [in] s int [in] prm1 int [in] prm2
Unserialize() void Public		Stream& [in] s int [in] prm1 int [in] prm2

ListenSocket

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {94D9F986-19C3-4808-9505-70C29D0737BE}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
socket SOCKET Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
whitelisted bool Public		<i>Default:</i>

Operations

Method	Notes	Parameters
ListenSocket() Public		SOCKET [in] socket bool [in] whitelisted

LocalServiceInfo**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {7315BF09-FDA9-4bf2-AF87-42A33E622E04}**Custom Properties**

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
nPort int Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
nScore int Public		<i>Default:</i>

LockedPageManager

Type: **Class** [LockedPageManagerBase](#)

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model [Keywords](#):

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {C86A6DE1-64EE-4239-AD69-7727D2396796}

Singleton class to keep track of locked (ie, non-swappable) memory pages, for use in std::allocator templates. Some implementations of the STL allocate memory in some constructors (i.e., see MSVC's vector<T> implementation where it allocates 1 byte of memory in the allocator.) Due to the unpredictable order of static initializers, we have to make sure the LockedPageManager instance exists before any other STL-based objects that use secure_allocator are created. So instead of having LockedPageManager also be static-initialized, it is created on demand.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public LockedPageManager	Private _instance LockedPageManager	
Generalization Source -> Destination	Public LockedPageManager	Public LockedPageManagerBa se	

Attributes

Attribute	Notes	Constraints and tags
_instance LockedPageManager* Private Static		<i>Default:</i>

Attribute	Notes	Constraints and tags
init_flag boost::once_flag Private Static		<i>Default:</i>

Operations

Method	Notes	Parameters
Static CreateInstance() void Private		
Static Instance() LockedPageManager& Public		
LockedPageManager() Private		

LockedPageManagerBase**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {E0940717-9424-4bc1-AF5A-291F22C95FE6}

Thread-safe class to keep track of locked (ie, non-swappable) memory pages. Memory locks do not stack, that is, pages which have been locked several times by calls to mlock() will be unlocked by a single call to munlock(). This can result in keying material ending up in swap when those functions are used naively. This class simulates stacking memory locks by keeping a counter per page.

@note By using a map from each page base address to lock count, this class is optimized for small objects that span up to a few pages, mostly smaller than a page. To support large allocations, something like an interval tree would be the preferred data structure.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public LockedPageManagerBa se	Private histogram Histogram	
Generalization Source -> Destination	Public LockedPageManager	Public LockedPageManagerBa se	

Connector	Source	Target	Notes

Attributes

Attribute	Notes	Constraints and tags
histogram Histogram Private		<i>Default:</i>
locker Locker Private		<i>Default:</i>
mutex boost::mutex Private		<i>Default:</i>
page_mask size_t Private		<i>Default:</i>
page_size size_t Private		<i>Default:</i>

Operations

Method	Notes	Parameters
GetLockedPageCount() int Public		
LockedPageManagerBas		size_t [in] page_size

Method	Notes	Parameters
e() Public		
~LockedPageManagerBase() Public		
LockRange() void Public		void* [in] p size_t [in] size
UnlockRange() void Public		void* [in] p size_t [in] size

Histogram

Type: Class **std::map<size_t, int>**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {CCB8CD21-D2A7-40b0-A31C-707C732C6886}

map of page base address to lock count

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public LockedPageManagerBase	Private histogram Histogram	

MacNotificationHandler

Type: Class **QObject**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {92A2EB91-C4C1-41bc-8896-61EC3FD8F818}

Macintosh-specific notification handler (supports UserNotificationCenter and Growl).

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
hasUserNotificationCenterSupport() bool Public		void [in] prm1
Static instance() MacNotificationHandler * Public		
sendAppleScript() void Public		QString& [in] script
showNotification() void Public		QString& [in] title QString& [in] text

MapCheckpoints

Type: **Class** **std::map<int, uint256>**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 G UID: {1F0F3445-2315-4fd0-87E2-11D75C93A5D7}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CCheckpointData	Public mapCheckpoints MapCheckpoints	

MemoryPageLocker

Type: **Class**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 G UID: {0D926EBF-76A8-416e-85A8-0F1720A03BF3}

OS-dependent memory page locking/unlocking. Defined as policy class to make stubbing for test possible.

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
Lock() bool Public		void* [in] addr size_t [in] len
Unlock() bool Public		void* [in] addr size_t [in] len

MutableTransactionSignatureChecker

Type: **Class** [TransactionSignatureChecker](#)
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 GUID: {483D6E1C-8A21-4ae3-97F3-743FD9FDBA53}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public MutableTransactionSignatureChecker	Private txTo CTransaction	
Generalization Source -> Destination	Public MutableTransactionSignatureChecker	Public TransactionSignatureChecker	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
txTo CTransaction Private Const		<i>Default:</i>

Operations

Method	Notes	Parameters
MutableTransactionSignatureChecker() Public		CMutableTransaction* [in] txToIn unsigned int [in] nInIn

Network

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {547BC573-8539-445c-9BDE-91F975179D9C}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
NET_UNROUTABLE Public		<i>Default:</i> 0

Attribute	Notes	Constraints and tags
NET_IPV4 Public		<i>Default:</i>
NET_IPV6 Public		<i>Default:</i>
NET_TOR Public		<i>Default:</i>
NET_MAX Public		<i>Default:</i>

NetworkStyle

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {6DFDC2FA-B25C-403a-AD57-EC2345997678}

Coin network-specific GUI style information

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
appIcon QIcon Private		<i>Default:</i>
appName QString Private		<i>Default:</i>
titleAddText QString Private		<i>Default:</i>
trayAndWindowIcon QIcon Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Const getAppIcon() QIcon & Public		
Const getAppName() QString & Public		
Const getTitleAddText() QString & Public		
Const getTrayAndWindowIcon() QIcon & Public		
Static Const instantiate() NetworkStyle * Public		QString& [in] networkId
NetworkStyle()	titleAddText needs to be const char* for tr()	QString& [in] appName

Method	Notes	Parameters
Private		int [in] iconColorHueShift int [in] iconColorSaturationReduction char* [in] titleAddText

Nodeld

Type: **Class** **int**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {8D31E100-CBA1-4a75-9AB1-BED6B37FA3FD}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CNodeStats	Public nodeid NodeId	
Association Source -> Destination	Public CNode	Public id NodeId	
Association Source -> Destination	Public COOrphanTx	Public fromPeer NodeId	

PaymentRequestPlus

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {62896E59-F120-4d51-9D3E-494B584B7965}

Wraps dumb protocol buffer paymentRequest with extra methods

Custom Properties

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public PaymentRequestPlus	Private details PaymentDetails	
Association Source -> Destination	Public PaymentRequestPlus	Private paymentRequest PaymentRequest	

Attributes

Attribute	Notes	Constraints and tags
details payments::PaymentDetails Private		<i>Default:</i>
paymentRequest payments::PaymentRequest Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Const getDetails() payments::PaymentDetails & Public		
getMerchant() bool Public		X509_STORE* [in] certStore QString& [in] merchant
getPayTo() QList<std::pair<CScript,C Amount> > Public		
IsInitialized() bool Public		
parse() bool		QByteArray& [in] data

Method	Notes	Parameters
Public		
PaymentRequestPlus() Public		
SerializeToString() bool Public		<code>std::string*</code> [in] output

PlatformStyle

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {A575B651-79CE-4bc7-AF9D-9E94290CF594}

Coin network-specific GUI style information

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
colorizeIcons bool Private		<i>Default:</i>
imagesOnButtons bool Private		<i>Default:</i>
name QString Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
singleColor QColor Private		<i>Default:</i>
textColor QColor Private		<i>Default:</i>
useExtraSpacing bool Private		<i>Default:</i>

Operations

Method	Notes	Parameters
getImagesOnButtons() bool Public		
Const getName() QString & Public		
getUseExtraSpacing() bool Public		
Static Const instantiate() PlatformStyle * Public		QString& [in] platformId
PlatformStyle() Private		QString& [in] name bool [in] imagesOnButtons bool [in] colorizeIcons bool [in] useExtraSpacing
SingleColor() QColor Public		
SingleColorIcon() QIcon Public		QString& [in] filename

Method	Notes	Parameters
SingleColorIcon() QIcon Public		QIcon& [in] icon
SingleColorImage() QImage Public		QString& [in] filename
TextColor() QColor Public		
TextColorIcon() QIcon Public		QString& [in] filename
TextColorIcon() QIcon Public		QIcon& [in] icon

ProxyCredentials

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {3F2749B7-F308-407a-A167-9902CCC7383A}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
password std::string Public		<i>Default:</i>
username std::string Public		<i>Default:</i>

RPCErrorCode

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {64528700-A5A6-4f04-B143-1AE627E242D3}

! Bitcoin RPC error codes

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
RPC_INVALID_REQUEST Public	! Standard JSON-RPC 2.0 errors	<i>Default:</i> -32600
RPC_METHOD_NOT_FOUND Public		<i>Default:</i> -32601
RPC_INVALID_PARAMS Public		<i>Default:</i> -32602
RPC_INTERNAL_ERROR Public		<i>Default:</i> -32603

Attribute	Notes	Constraints and tags
RPC_PARSE_ERROR Public		<i>Default:</i> -32700
RPC_MISC_ERROR Public	! std::exception thrown in command handling	<i>Default:</i> -1
RPC_FORBIDDEN_BY_SAFE_MODE Public	! Server is in safe mode, and command is not allowed in safe mode	<i>Default:</i> -2
RPC_TYPE_ERROR Public	! Unexpected type was passed as parameter	<i>Default:</i> -3
RPC_INVALID_ADDRESS_OR_KEY Public	! Invalid address or key	<i>Default:</i> -5
RPC_OUT_OF_MEMORY Public	! Ran out of memory during operation	<i>Default:</i> -7

Attribute	Notes	Constraints and tags
RPC_INVALID_PARA METER Public	! Invalid, missing or duplicate parameter	<i>Default:</i> -8
RPC_DATABASE_ERR OR Public	! Database error	<i>Default:</i> -20
RPC_DESERIALIZATI ON_ERROR Public	! Error parsing or validating structure in raw format	<i>Default:</i> -22
RPC_VERIFY_ERROR Public	! General error during transaction or block submission	<i>Default:</i> -25
RPC_VERIFY_REJECT ED Public	! Transaction or block was rejected by network rules	<i>Default:</i> -26
RPC_VERIFY_ALREA DY_IN_CHAIN Public	! Transaction already in chain	<i>Default:</i> -27

Attribute	Notes	Constraints and tags
RPC_IN_WARMUP Public	! Client still warming up	<i>Default:</i> -28
RPC_TRANSACTION_ERROR Public	! Aliases for backward compatibility	<i>Default:</i> RPC_VERIFY_ERROR
RPC_TRANSACTION_REJECTED Public		<i>Default:</i> RPC_VERIFY_REJECTED
RPC_TRANSACTION_ALREADY_IN_CHAIN Public		<i>Default:</i> RPC_VERIFY_ALREADY_IN_CHAIN
RPC_CLIENT_NOT_CONNECTED Public	! Bitcoin is not connected	<i>Default:</i> -9
RPC_CLIENT_IN_INITIAL_DOWNLOAD Public	! Still downloading initial blocks	<i>Default:</i> -10

Attribute	Notes	Constraints and tags
RPC_CLIENT_NODE_A LREADY_ADDED Public	! Node is already added	<i>Default:</i> -23
RPC_CLIENT_NODE_N OT_ADDED Public	! Node has not been added before	<i>Default:</i> -24
RPC_CLIENT_NODE_N OT_CONNECTED Public	! Node to disconnect not found in connected nodes	<i>Default:</i> -29
RPC_CLIENT_INVALID_IP_OR_SUBNET Public	! Invalid IP/Subnet	<i>Default:</i> -30
RPC_WALLET_ERROR Public	! Unspecified problem with wallet (key not found etc.)	<i>Default:</i> -4
RPC_WALLET_INSUFFICIENT_FUNDS Public	! Not enough funds in wallet or account	<i>Default:</i> -6

Attribute	Notes	Constraints and tags
RPC_WALLET_INVALID_ACCOUNT_NAME Public	! Invalid account name	<i>Default:</i> -11
RPC_WALLET_KEYPOOL_RAN_OUT Public	! Keypool ran out, call keypoolrefill first	<i>Default:</i> -12
RPC_WALLET_UNLOCK_NEEDED Public	! Enter the wallet passphrase with walletpassphrase first	<i>Default:</i> -13
RPC_WALLET_PASSPHRASE_INCORRECT Public	! The wallet passphrase entered was incorrect	<i>Default:</i> -14
RPC_WALLET_WRONG_ENC_STATE Public	! Command given in wrong wallet encryption state (encrypting an encrypted wallet etc.)	<i>Default:</i> -15
RPC_WALLET_ENCRYPTION_FAILED Public	! Failed to encrypt the wallet	<i>Default:</i> -16

Attribute	Notes	Constraints and tags
RPC_WALLET_ALREADY_UNLOCKED Public	! Wallet is already unlocked	<i>Default:</i> -17

RecentRequestEntry

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {6735096F-7648-4481-B8A6-923BDB188A39}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public RecentRequestEntry	Public id int64_t	

Attributes

Attribute	Notes	Constraints and tags
ADD_SERIALIZE_METHODS int Public		<i>Default:</i>
CURRENT_VERSION int Public Static Const		<i>Default:</i> 1

Attribute	Notes	Constraints and tags
date QDateTime Public		<i>Default:</i>
id int64_t Public		<i>Default:</i>
nVersion int Public		<i>Default:</i>
recipient SendCoinsRecipient Public		<i>Default:</i>

Operations

Method	Notes	Parameters
RecentRequestEntry() Public		
SerializationOp() void Public		Stream& [in] s Operation [in] ser_action int [in] nType int [in] nVersion

RecentRequestEntryLessThan**Type:****Class****Status:**

Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {BF23DD79-E5AF-4cf3-A930-A512FE0EEDA7}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
column int Private		<i>Default:</i>
order Qt::SortOrder Private		<i>Default:</i>

Operations

Method	Notes	Parameters
operator() bool Public		RecentRequestEntry& [in] left RecentRequestEntry& [in] right
RecentRequestEntryLessThan() Public		int [in] nColumn Qt::SortOrder [in] fOrder

SSLIOSStreamDevice

Type: **Class** **boost::iostreams::device**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {2830148C-CB09-4ddd-BAC8-EC2766458268}

IOStream device that speaks SSL but can also speak non-SSL

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
fNeedHandshake bool Private		<i>Default:</i>
fUseSSL bool Private		<i>Default:</i>
stream boost::asio::ssl::stream<typename Protocol::socket>& Private		<i>Default:</i>

Operations

Method	Notes	Parameters
connect() bool Public		std::string& [in] server std::string& [in] port
handshake() void Public		boost::asio::ssl::stream_base::handshake_type [in] role
read() std::streamsize Public		char* [in] s std::streamsize [in] n

Method	Notes	Parameters
SSLIOStreamDevice() Public		boost::asio::ssl::stream<typename Protocol::socket>& [in] streamIn bool [in] fUseSSLIn
write() std::streamsize Public		char* [in] s std::streamsize [in] n

SSLVerifyError

Type: [Class](#) [std::runtime_error](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords](#):
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {C9AB96CB-86F7-4fe3-A47E-EA6CD6178B26}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
SSLVerifyError() Public		std::string [in] err

ScriptError_t

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords](#):
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {C6F50DA3-78CF-4263-925B-EF735A793F98}

Custom Properties

- isActive = False

Tagged Values

- `typedef = true.`
- `typeSynonyms = ScriptError.`

Attributes

Attribute	Notes	Constraints and tags
SCRIPT_ERR_OK Public		<i>Default:</i> 0
SCRIPT_ERR_UNKNO WN_ERROR Public		<i>Default:</i>
SCRIPT_ERR_EVAL_F ALSE Public		<i>Default:</i>
SCRIPT_ERR_OP_RET URN Public		<i>Default:</i>
SCRIPT_ERR_SCRIPT_SIZE Public	Max sizes	<i>Default:</i>

Attribute	Notes	Constraints and tags
SCRIPT_ERR_PUSH_SIZE Public		<i>Default:</i>
SCRIPT_ERR_OP_COUNT Public		<i>Default:</i>
SCRIPT_ERR_STACK_SIZE Public		<i>Default:</i>
SCRIPT_ERR_SIG_COUNT Public		<i>Default:</i>
SCRIPT_ERR_PUBKEY_COUNT Public		<i>Default:</i>
SCRIPT_ERR_VERIFY Public	Failed verify operations	<i>Default:</i>

Attribute	Notes	Constraints and tags
SCRIPT_ERR_EQUALVERIFY Public		<i>Default:</i>
SCRIPT_ERR_CHECKMULTISIGVERIFY Public		<i>Default:</i>
SCRIPT_ERR_CHECKSIGVERIFY Public		<i>Default:</i>
SCRIPT_ERR_NUMEQUALVERIFY Public		<i>Default:</i>
SCRIPT_ERR_BAD_OPCODE Public	Logical/Format/Canonical errors	<i>Default:</i>
SCRIPT_ERR_DISABLED_OPCODE Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
SCRIPT_ERR_INVALID_STACK_OPERATION Public		<i>Default:</i>
SCRIPT_ERR_INVALID_ALTSTACK_OPERATION Public		<i>Default:</i>
SCRIPT_ERR_UNBALANCED_CONDITIONAL Public		<i>Default:</i>
SCRIPT_ERR_NEGATIVE_LOCKTIME Public	OP_CHECKLOCKTIMEVERIFY	<i>Default:</i>
SCRIPT_ERR_UNSATISFIED_LOCKTIME Public		<i>Default:</i>
SCRIPT_ERR_SIG_HASHTYPE Public	BIP62	<i>Default:</i>

Attribute	Notes	Constraints and tags
SCRIPT_ERR_SIG_DEF Public		<i>Default:</i>
SCRIPT_ERR_MINIMALDATA Public		<i>Default:</i>
SCRIPT_ERR_SIG_PUSHONLY Public		<i>Default:</i>
SCRIPT_ERR_SIG_HIGH_S Public		<i>Default:</i>
SCRIPT_ERR_SIG_NULLDUMMY Public		<i>Default:</i>
SCRIPT_ERR_PUBKEY_TYPE Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
SCRIPT_ERR_CLEANS_TACK Public		<i>Default:</i>
SCRIPT_ERR_DISCOURAGE_UPGRADABLE_NOPS Public	softfork safeness	<i>Default:</i>
SCRIPT_ERR_ERROR_COUNT Public		<i>Default:</i>

ScriptMap

Type: Class std::map<CScriptID, CScript >
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {B81BCCE3-8365-46e8-B7C8-7F9CCD5F1D90}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBasicKeyStore	Protected mapScripts ScriptMap	

SecureString

Type: **Class** **`std::basic_string<char, std::char_traits<char>, secure_allocator<char>`**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {41F07EF1-3168-42fd-9F6E-BCECBA124AEB}

This is exactly like `std::string`, but with a custom allocator.

Custom Properties

- `isActive` = False

SeedSpec6

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {5CE0F0A6-D87F-4c42-9A86-8937F5B86915}

Custom Properties

- `isActive` = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public SeedSpec6	Public addr uint8_t	
Association Source -> Destination	Public SeedSpec6	Public port uint16_t	

Attributes

Attribute	Notes	Constraints and tags
addr uint8_t Public Collection		<i>Default:</i>

Attribute	Notes	Constraints and tags
port uint16_t Public		<i>Default:</i>

TestChain100Setup

Type: Class **TestingSetup**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {58170BEC-E06C-4a95-AA25-9BB5F173396B}

Testing fixture that pre-creates a 100-block REGTEST-mode block chain

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TestChain100Setup	Public coinbaseKey CKey	
Generalization Source -> Destination	Public TestChain100Setup	Public TestingSetup	

Attributes

Attribute	Notes	Constraints and tags
coinbaseKey CKey Public	private/public key needed to spend coinbase transactions	<i>Default:</i>

Attribute	Notes	Constraints and tags
coinbaseTxns std::vector<CTransaction> Public	For convenience, coinbase transactions	<i>Default:</i>

Operations

Method	Notes	Parameters
CreateAndProcessBlock() CBlock Public	Create a new block with just given transactions, coinbase paying to scriptPubKey, and try to add it to the current chain.	std::vector<CMutableTransaction>& [in] txns CScript& [in] scriptPubKey
TestChain100Setup() Public		
~TestChain100Setup() Public		

TestingSetup

Type: **Class** BasicTestingSetup
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 G UID: {A10AEAE7-0EA2-43c0-AFB3-22F4823A2F39}

Testing setup that configures a complete environment. Included are data directory, coins database, script check threads and wallet (if enabled) setup.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TestingSetup	Public pcoinsdbview CCoinsViewDB	
Generalization Source -> Destination	Public TestingSetup	Public BasicTestingSetup	
Generalization Source -> Destination	Public TestChain100Setup	Public TestingSetup	

Attributes

Attribute	Notes	Constraints and tags
pathTemp boost::filesystem::path Public		<i>Default:</i>
pcoinsdbview CCoinsViewDB* Public		<i>Default:</i>
threadGroup boost::thread_group Public		<i>Default:</i>

Operations

Method	Notes	Parameters
TestingSetup() Public		CBaseChainParams::Network [in] network
~TestingSetup() Public		

TransactionDesc

Type: **Class** **QObject**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A803AA75-C076-4546-90FB-A8177E504922}

Provide a human-readable extended HTML description of a transaction.

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
Static FormatTxStatus() QString Private		CWalletTx& [in] wtx
Static toHTML() QString Public		CWallet* [in] wallet CWalletTx& [in] wtx TransactionRecord* [in] rec int [in] unit
TransactionDesc() Private		

TransactionFilterProxy

Type: **Class** **QSortFilterProxyModel**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 GUID: {655F4383-DDFC-4662-BAC3-D72D6CF75C1E}

Filter the transaction list according to pre-specified rules.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TransactionFilterProxy	Private minAmount CAmount	
Association Source -> Destination	Public TransactionFilterProxy	Private watchOnlyFilter WatchOnlyFilter	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
addrPrefix QString Private		<i>Default:</i>
ALL_TYPES quint32 Public Static Const	Type filter bit field (all types)	<i>Default:</i> 0xFFFFFFFF
dateFrom QDateTime Private		<i>Default:</i>
dateTo QDateTime Private		<i>Default:</i>
limitRows int Private		<i>Default:</i>
MAX_DATE QDateTime Public Static Const	Last date that can be represented (far in the future)	<i>Default:</i>

Attribute	Notes	Constraints and tags
MIN_DATE QDateTime Public Static Const	Earliest date that can be represented (far in the past)	<i>Default:</i>
minAmount CAmount Private		<i>Default:</i>
showInactive bool Private		<i>Default:</i>
typeFilter quint32 Private		<i>Default:</i>
watchOnlyFilter WatchOnlyFilter Private		<i>Default:</i>

Operations

Method	Notes	Parameters
filterAcceptsRow() bool Protected		int [in] source_row QModelIndex& [in] source_parent
rowCount() int Public		QModelIndex& [in] parent
setAddressPrefix() void		QString& [in] addrPrefix

Method	Notes	Parameters
Public		
setDateRange() void Public		QDateTime& [in] from QDateTime& [in] to
setLimit() void Public		int [in] limit
setMinAmount() void Public		CAmount& [in] minimum
 setShowInactive() void Public		bool [in] showInactive
setTypeFilter() void Public		quint32 [in] modes
setWatchOnlyFilter() void Public		WatchOnlyFilter [in] filter
TransactionFilterProxy() Public		QObject* [in] parent
Static TYPE() quint32 Public		int [in] type

WatchOnlyFilter

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {BD24F3CF-79DD-4089-8A83-30DF71CD468B}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TransactionFilterProxy	Private watchOnlyFilter WatchOnlyFilter	

Attributes

Attribute	Notes	Constraints and tags
WatchOnlyFilter_All Public		<i>Default:</i>
WatchOnlyFilter_Yes Public		<i>Default:</i>
WatchOnlyFilter_No Public		<i>Default:</i>

TransactionRecord

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A7768ED2-32B2-4f2f-B00A-43D72DECD7C2}

UI model for a transaction. A core transaction can be represented by multiple UI transactions if it has multiple outputs.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TransactionRecord	Public credit CAmount	
Association Source -> Destination	Public TransactionRecord	Public debit CAmount	

Connector	Source	Target	Notes
Association Source -> Destination	Public TransactionRecord	Public hash uint256	
Association Source -> Destination	Public TransactionRecord	Public status TransactionStatus	
Association Source -> Destination	Public TransactionRecord	Public type Type	

Attributes

Attribute	Notes	Constraints and tags
address std::string Public		<i>Default:</i>
credit CAmount Public		<i>Default:</i>
debit CAmount Public		<i>Default:</i>
hash uint256 Public	@name Immutable transaction attributes @{	<i>Default:</i>
idx int Public	Subtransaction index, for sort key	<i>Default:</i>

Attribute	Notes	Constraints and tags
involvesWatchAddress bool Public	Whether the transaction was sent/received with a watch-only address	<i>Default:</i>
RecommendedNumConfirms int Public Static Const	Number of confirmation recommended for accepting a transaction	<i>Default:</i> 6
status TransactionStatus Public	Status: can change with block chain update	<i>Default:</i>
time qint64 Public		<i>Default:</i>
type Type Public		<i>Default:</i>

Operations

Method	Notes	Parameters
Static decomposeTransaction() QList<TransactionRecord> Public	Decompose CWallet transaction to model transaction records.	CWallet* [in] wallet CWalletTx& [in] wtx
Static formatSubTxId() QString Public		uint256& [in] hash int [in] vout

Method	Notes	Parameters
getTxID() QString Public		
Static showTransaction() bool Public	Return positive answer if transaction should be shown in list.	CWalletTx& [in] wtx
statusUpdateNeeded() bool Public		
TransactionRecord() Public		
TransactionRecord() Public		uint256 [in] hash qint64 [in] time
TransactionRecord() Public		uint256 [in] hash qint64 [in] time Type [in] type std::string& [in] address CAmount& [in] debit CAmount& [in] credit
updateStatus() void Public		CWalletTx& [in] wtx

Type

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {E5245805-496F-4620-8D5C-594C5B654FCF}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TransactionRecord	Public type Type	

Connector	Source	Target	Notes

Attributes

Attribute	Notes	Constraints and tags
Other Public		<i>Default:</i>
Generated Public		<i>Default:</i>
SendToAddress Public		<i>Default:</i>
SendToOther Public		<i>Default:</i>
RecvWithAddress Public		<i>Default:</i>
RecvFromOther Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
SendToSelf Public		<i>Default:</i>

TransactionSignatureChecker

Type: Class **BaseSignatureChecker**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {5A9D6E85-2851-4d83-B118-C2B4F51E3BA9}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TransactionSignatureCh ecker	Private txTo CTransaction	
Generalization Source -> Destination	Public TransactionSignatureCh ecker	Public BaseSignatureChecker	
Generalization Source -> Destination	Public MutableTransactionSig natureChecker	Public TransactionSignatureCh ecker	
Generalization Source -> Destination	Public CachingTransactionSig natureChecker	Public TransactionSignatureCh ecker	
Association Source -> Destination	Public TransactionSignatureCr eator	Private checker TransactionSignatureCh ecker	

Attributes

Attribute	Notes	Constraints and tags
nIn unsigned int Private		<i>Default:</i>
txTo CTransaction* Private Const		<i>Default:</i>

Operations

Method	Notes	Parameters
CheckLockTime() bool Public		CScriptNum& [in] nLockTime
CheckSig() bool Public		std::vector<unsigned char>& [in] scriptSig std::vector<unsigned char>& [in] vchPubKey CScript& [in] scriptCode
TransactionSignatureChecker() Public		CTransaction* [in] txToIn unsigned int [in] nInIn
VerifySignature() bool Protected		std::vector<unsigned char>& [in] vchSig CPubKey& [in] vchPubKey uint256& [in] sighash

TransactionSignatureCreator

Type: **Class** **BaseSignatureCreator**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {95C4765C-E3F1-4845-8886-0362AF44CC1D}

A signature creator for transactions.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TransactionSignatureCreator	Private checker TransactionSignatureChecker	
Association Source -> Destination	Public TransactionSignatureCreator	Private txTo CTransaction	
Generalization Source -> Destination	Public TransactionSignatureCreator	Public BaseSignatureCreator	

Attributes

Attribute	Notes	Constraints and tags
checker TransactionSignatureChecker Private Const		<i>Default:</i>
nHashType int Private		<i>Default:</i>
nIn unsigned int Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
txTo CTransaction*	Private Const	<i>Default:</i>

Operations

Method	Notes	Parameters
Const Checker() BaseSignatureChecker& Public		
CreateSig() bool Public		std::vector<unsigned char>& [in] vchSig CKeyID& [in] keyid CScript& [in] scriptCode
TransactionSignatureCreator() Public		CKeyStore* [in] keystoreIn CTransaction* [in] txToIn unsigned int [in] nInIn int [in] nHashTypeIn

TransactionStatus

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {59072820-00B1-4fde-B0C2-7ECD8ED30CFF}

UI model for transaction status. The transaction status is the part of a transaction that will change over time.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association	Public	Public status	

Connector	Source	Target	Notes
Source -> Destination	TransactionStatus	Status	
Association Source -> Destination	Public TransactionRecord	Public status TransactionStatus	

Attributes

Attribute	Notes	Constraints and tags
countsForBalance bool Public	Transaction counts towards available balance	<i>Default:</i>
cur_num_blocks int Public	Current number of blocks (to know whether cached status is still valid)	<i>Default:</i>
depth qint64 Public		<i>Default:</i>
matures_in int Public	@name Generated (mined) transactions @{	<i>Default:</i>
open_for qint64 Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
sortKey std::string Public	Sorting key based on status	<i>Default:</i>
status Status Public	@name Reported status @{	<i>Default:</i>

Operations

Method	Notes	Parameters
TransactionStatus() Public		

Status

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {C8E819FC-18D4-4f29-A3C4-04FCE36D121F}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TransactionStatus	Public status Status	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
Confirmed Public	< Have 6 or more confirmations (normal tx) or fully mature (mined tx)	<i>Default:</i>
OpenUntilDate Public	< Transaction not yet final, waiting for date	<i>Default:</i>
OpenUntilBlock Public	< Transaction not yet final, waiting for block	<i>Default:</i>
Offline Public	< Not sent to any other nodes	<i>Default:</i>
Unconfirmed Public	< Not yet mined into a block	<i>Default:</i>
Confirming Public	< Confirmed, but waiting for the recommended number of confirmations	<i>Default:</i>

Attribute	Notes	Constraints and tags
Conflicted Public	< Conflicts with other transaction or mempool	<i>Default:</i>
Immature Public	< Mined but waiting for maturity	<i>Default:</i>
MaturesWarning Public	< Transaction will likely not mature because no nodes have confirmed	<i>Default:</i>
NotAccepted Public	< Mined but not accepted	<i>Default:</i>

TxConfirmStats

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords](#):
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {CFD4AEEE-3DE7-4b0e-9E7A-E4F1A3CE96B2}

We will instantiate two instances of this class, one to track transactions that were included in a block due to fee, and one for tx's included due to priority. We will lump transactions into a bucket according to their approximate fee or priority and then track how long it took for those txs to be included in a block. The tracking of unconfirmed (mempool) transactions is completely independent of the historical tracking of transactions that have been confirmed in a block.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBlockPolicyEstimator	Private feeStats TxConfirmStats	
Association Source -> Destination	Public CBlockPolicyEstimator	Private priStats TxConfirmStats	
Association Source -> Destination	Public TxStatsInfo	Public stats TxConfirmStats	

Attributes

Attribute	Notes	Constraints and tags
avg std::vector<double> Private	Sum the total priority/fee of all tx's in each bucket Track the historical moving average of this total over blocks	<i>Default:</i>
bucketMap std::map<double, unsigned int> Private	Map of bucket upper-bound to index into all vectors by bucket	<i>Default:</i>
buckets std::vector<double> Private	The upper-bound of the range for the bucket (inclusive)	<i>Default:</i>
confAvg std::vector<std::vector<double> > Private	Count the total # of txs confirmed within Y blocks in each bucket Track the historical moving average of these totals over blocks	<i>Default:</i>

Attribute	Notes	Constraints and tags
curBlockConf std::vector<std::vector<int>> Private	curBlockConf[Y][X]	<i>Default:</i>
curBlockTxCt std::vector<int> Private	and calculate the total for the current block to update the moving average	<i>Default:</i>
curBlockVal std::vector<double> Private	and calculate the total for the current block to update the moving average	<i>Default:</i>
dataTypeString std::string Private	Combine the conf counts with tx counts to calculate the confirmation % for each Y,X Combine the total value with the tx counts to calculate the avg fee/priority per bucket	<i>Default:</i>
decay double Private		<i>Default:</i>
oldUnconfTxs std::vector<int> Private	unconfTxs[Y][X] transactions still unconfirmed after MAX_CONFIRMMS for each bucket	<i>Default:</i>

Attribute	Notes	Constraints and tags
txCtAvg std::vector<double> Private	For each bucket X: Count the total # of txs in each bucket Track the historical moving average of this total over blocks	<i>Default:</i>
unconfTxs std::vector<std::vector<int>> Private	Mempool counts of outstanding transactions For each bucket X, track the number of transactions in the mempool that are unconfirmed for each possible confirmation value Y	<i>Default:</i>

Operations

Method	Notes	Parameters
ClearCurrent() void Public	Zero out the data for the current block	unsigned int [in] nBlockHeight
EstimateMedianVal() double Public	returns -1 on error conditions	int [in] confTarget double [in] sufficientTxVal double [in] minSuccess bool [in] requireGreater unsigned int [in] nBlockHeight
GetMaxConfirms() unsigned int Public		
Initialize() void Public		std::vector<double>& [in] defaultBuckets unsigned int [in] maxConfirms double [in] decay std::string [in] dataTypeString
NewTx() unsigned int Public		unsigned int [in] nBlockHeight double [in] val
Read() void		CAutoFile& [in] filein

Method	Notes	Parameters
Public		
Record() void Public		int [in] blocksToConfirm double [in] val
removeTx() void Public		unsigned int [in] entryHeight unsigned int [in] nBestSeenHeight unsigned int [in] bucketIndex
UpdateMovingAverages() void Public		
Write() void Public		CAutoFile& [in] fileout

TxInputStream

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {EB0A182E-A74E-4528-9F9A-1EE48879B548}

A class that deserializes a single CTransaction one time.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
m_data unsigned char* Private Const		<i>Default:</i>

Attribute	Notes	Constraints and tags
m_remaining size_t Private		<i>Default:</i>
m_type int Private Const		<i>Default:</i>
m_version int Private Const		<i>Default:</i>

Operations

Method	Notes	Parameters
operator>> () TxInputStream& Public		T& [in] obj
read() TxInputStream& Public		char* [in] pch size_t [in] nSize
TxInputStream() Public		int [in] nTypeIn int [in] nVersionIn unsigned char* [in] txTo size_t [in] txToLen

TxPriority**Type:****Class** **boost::tuple<double, CFeeRate, const CTransaction*>****Status:**

Proposed. Version 1.0. Phase 1.0.

Package:

Class Model

Keywords:**Detail:**

Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {A31B2A2C-8A53-46f2-84D8-C996601DF9EC}

We want to sort transactions by priority and fee rate, so:

Custom Properties

- isActive = False

TxPriorityCompare

Type: Class

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {10B77EAA-F342-400c-85EA-18CB62F98315}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
byFee bool Private		<i>Default:</i>

Operations

Method	Notes	Parameters
operator()() bool Public		TxPriority& [in] a TxPriority& [in] b
TxPriorityCompare() Public		bool [in] _byFee

UniValue

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {34D203BE-9740-4a9e-AE85-F6DA042AFB50}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public UniValue	Private typ VType	

Attributes

Attribute	Notes	Constraints and tags
keys std::vector<std::string> Private		<i>Default:</i>
typ UniValue::VType Private		<i>Default:</i>
val std::string Private	numbers are stored as C++ strings	<i>Default:</i>

Attribute	Notes	Constraints and tags
values std::vector<UniValue> Private		<i>Default:</i>

Operations

Method	Notes	Parameters
checkObject() bool Public		std::map<std::string,UniValue::VType>& [in] memberTypes
clear() void Public		
empty() bool Public		
exists() bool Public		std::string& [in] key
Const find_value() UniValue& Public		UniValue& [in] obj std::string& [in] name
findKey() int Private		std::string& [in] key
Const get_array() UniValue& Public		
get_bool() bool Public		
get_int() int Public		
get_int64() int64_t Public		
Const get_obj() UniValue& Public		
get_real() double Public		
get_str() std::string Public		
getBool() bool Public		
getKeys() std::vector<std::string> Public		
getType() enum VType Public		

Method	Notes	Parameters
Const getValStr() std::string& Public		
getValues() std::vector<UniValue> Public		
isArray() bool Public		
isBool() bool Public		
isFalse() bool Public		
isNull() bool Public		
isNum() bool Public		
isObject() bool Public		
isStr() bool Public		
isTrue() bool Public		
Const operator[]() UniValue& Public		std::string& [in] key
Const operator[]() UniValue& Public		unsigned int [in] index
push_back() bool Public		UniValue& [in] val
push_back() bool Public		std::string& [in] val_
push_back() bool Public		char* [in] val_
push_back() bool Public		std::pair<std::string,UniValue> [in] pair
push_backV() bool Public		std::vector<UniValue>& [in] vec
pushKV() bool Public		std::string& [in] key UniValue& [in] val
pushKV() bool Public		std::string& [in] key std::string& [in] val

Method	Notes	Parameters
pushKV() bool Public		std::string& [in] key char* [in] val_
pushKV() bool Public		std::string& [in] key int64_t [in] val
pushKV() bool Public		std::string& [in] key uint64_t [in] val
pushKV() bool Public		std::string& [in] key int [in] val
pushKV() bool Public		std::string& [in] key double [in] val
pushKVs() bool Public		UniValue& [in] obj
read() bool Public		char* [in] raw
read() bool Public		std::string& [in] rawStr
setArray() bool Public		
setBool() bool Public		bool [in] val
setFloat() bool Public		double [in] val
setInt() bool Public		uint64_t [in] val
setInt() bool Public		int64_t [in] val
setInt() bool Public		int [in] val
setNull() bool Public		
setNumStr() bool		std::string& [in] val

Method	Notes	Parameters
Public		
setObject() bool Public		
setStr() bool Public		std::string& [in] val
size() size_t Public		
type() enum VType Public		
UniValue() Public		
UniValue() Public		UniValue::VType [in] initialType std::string& [in] initialStr
UniValue() Public		uint64_t [in] val_
UniValue() Public		int64_t [in] val_
UniValue() Public		bool [in] val_
UniValue() Public		int [in] val_
UniValue() Public		double [in] val_
UniValue() Public		std::string& [in] val_
UniValue() Public		char* [in] val_
~UniValue() Public		
write() std::string Public		unsigned int [in] prettyIndent unsigned int [in] indentLevel
writeArray() void Private		unsigned int [in] prettyIndent unsigned int [in] indentLevel std::string& [in] s
writeObject() void		unsigned int [in] prettyIndent

Method	Notes	Parameters
Private		unsigned int [in] indentLevel std::string& [in] s

VType

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {3525F805-E398-4abd-8B71-2FA16C7212D5}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public UniValue	Private typ VType	

Attributes

Attribute	Notes	Constraints and tags
VNULL Public		<i>Default:</i>
VOBJ Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
VARR Public		<i>Default:</i>
VSTR Public		<i>Default:</i>
VNUM Public		<i>Default:</i>
VBOOL Public		<i>Default:</i>

WalletFeature

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords](#):
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {6FFA7D86-3E18-48a0-8DB3-CFDE8DD910F0}

(client) version numbers for particular wallet features

[Custom Properties](#)

- isActive = False

[Attributes](#)

Attribute	Notes	Constraints and tags
FEATURE_BASE Public	the earliest version new wallets supports (only useful for getinfo's clientversion output)	<i>Default:</i> 10500
FEATURE_WALLETCRYPT Public	wallet encryption	<i>Default:</i> 40000
FEATURE_COMPRPUBKEY Public	compressed public keys	<i>Default:</i> 60000
FEATURE_LATEST Public		<i>Default:</i> 60000

WalletModelTransaction

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A688799B-8A5C-4618-B0E0-09D836552E09}

Data model for a walletmodel transaction.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
-----------	--------	--------	-------

Connector	Source	Target	Notes
Association Source -> Destination	Public WalletModelTransactio n	Private fee CAmount	
Association Source -> Destination	Public WalletModelTransactio n	Private keyChange CReserveKey	
Association Source -> Destination	Public WalletModelTransactio n	Private walletTransaction CWalletTx	

Attributes

Attribute	Notes	Constraints and tags
fee CAmount Private		<i>Default:</i>
keyChange CReserveKey* Private		<i>Default:</i>
recipients QList<SendCoinsRecipient > Private		<i>Default:</i>
walletTransaction CWalletTx* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
--------	-------	------------

Method	Notes	Parameters
getPossibleKeyChange() CReserveKey * Public		
getRecipients() QList<SendCoinsRecipient > Public		
getTotalTransactionAmo unt() CAmount Public		
getTransaction() CWalletTx * Public		
getTransactionFee() CAmount Public		
getTransactionSize() unsigned int Public		
newPossibleKeyChange() void Public		CWallet* [in] wallet
reassignAmounts() void Public		int [in] nChangePosRet
setTransactionFee() void Public		CAmount& [in] newFee
WalletModelTransaction() Public		QList<SendCoinsRecipient>& [in] recipients
~WalletModelTransactio n() Public		

WatchKeyMap

Type: Class std::map<CKeyID, CPubKey>
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {E0CB88D3-B960-41b2-B431-996DF4C258DD}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBasicKeyStore	Protected mapWatchKeys WatchKeyMap	

WatchOnlySet

Type: **Class** **std::set<CScript>**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 G UID: {F2828555-0058-43cf-911A-649C3643E7A6}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CBasicKeyStore	Protected setWatchOnly WatchOnlySet	

WinShutdownMonitor

Type: **Class** **QAbstractNativeEventFilter**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 G UID: {7C3340F5-3D35-4c72-9338-7933B2695839}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
nativeEventFilter() bool Public	If we don't want a message to be processed by Qt, return true and set result to the value that the window procedure should return. Otherwise return false.	QByteArray& [in] eventType void* [in] pMessage

Method	Notes	Parameters
		long* [in] pnResult
Static registerShutdownBlockReason() void Public		QString& [in] strReason HWND& [in] mainWinId

arith_uint256

Type: **Class** **base_uint**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {87A8EAFE-EB1D-4756-8655-4D963A233ECF}

256-bit unsigned big integer.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public arith_uint256	Public base_uint	
Association Source -> Destination	Public CBlockIndex	Public nChainWork arith_uint256	

Operations

Method	Notes	Parameters
arith_uint256() Public		
arith_uint256() Public		base_uint<256>& [in] b
arith_uint256() Public		uint64_t [in] b
arith_uint256() Public		std::string& [in] str
ArithToUint256() uint256 Public		arith_uint256& [in] prm1
GetCompact() uint32_t		bool [in] fNegative

Method	Notes	Parameters
Public		
SetCompact() arith_uint256& Public	This implementation directly uses shifts instead of going through an intermediate MPI representation.	uint32_t [in] nCompact bool* [in] pfNegative bool* [in] pfOverflow
UintToArith256() arith_uint256 Public		uint256& [in] prm1

banmap_t

Type: [Class](#) [std::map<CSubNet, CBanEntry>](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords:](#)
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {2AF1E3E1-BEA9-4440-9E8E-6F0FB7787864}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CNode	Protected setBanned banmap_t	

base_blob

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords:](#)
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {8663257B-7E9E-4a1f-B456-01D7606D9A17}

Template base class for fixed-sized opaque blobs.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public base_blob	Protected data uint8_t	
Generalization Source -> Destination	Public uint160	Public base_blob	
Generalization Source -> Destination	Public uint256	Public base_blob	

Attributes

Attribute	Notes	Constraints and tags
data uint8_t Protected Collection		<i>Default:</i>

Operations

Method	Notes	Parameters
base_blob() Public		
base_blob() Public		std::vector<unsigned char>& [in] vch
begin() unsigned char* Public		
Const begin() unsigned char* Public		
end() unsigned char* Public		
Const end() unsigned char* Public		
GetHex() std::string Public		
GetSerializeSize() unsigned int Public		int [in] nType int [in] nVersion
IsNull() bool Public		
operator!=() bool Public		base_blob& [in] a

Method	Notes	Parameters
		base_blob& [in] b
operator<() bool Public		base_blob& [in] a base_blob& [in] b
operator==() bool Public		base_blob& [in] a base_blob& [in] b
Serialize() void Public		Stream& [in] s int [in] nType int [in] nVersion
SetHex() void Public		char* [in] psz
SetHex() void Public		std::string& [in] str
SetNull() void Public		
size() unsigned int Public		
ToString() std::string Public		
Unserialize() void Public		Stream& [in] s int [in] nType int [in] nVersion

<anonymous>

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {EB8F3A99-DD88-44cd-8C16-8A6F018C1A97}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
WIDTH Protected		<i>Default:</i> BITS/8

base_uint**Type:** [Class](#)**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {64D9D7CD-C154-459f-86EF-4F3795A4C8EF}

Template base class for unsigned big integers.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public base_uint	Protected pn uint32_t	
Generalization Source -> Destination	Public arith_uint256	Public base_uint	

Attributes

Attribute	Notes	Constraints and tags
pn uint32_t Protected Collection		<i>Default:</i>

Operations

Method	Notes	Parameters
base_uint() Public		
base_uint() Public		base_uint& [in] b
base_uint() Public		uint64_t [in] b
base_uint() Public		std::string& [in] str
bits() unsigned int Public		
CompareTo() int Public		base_uint& [in] b
EqualTo() bool Public		uint64_t [in] b
getdouble() double Public		
GetHex() std::string Public		
GetLow64() uint64_t Public		
Const operator-() base_uint Public		
Const operator-() base_uint Public		base_uint& [in] a base_uint& [in] b
operator--() base_uint& Public		
Const operator--() base_uint Public		int [in] prm1
operator-=() base_uint& Public		base_uint& [in] b
operator-=() base_uint& Public		uint64_t [in] b64
operator!() bool Public		
operator!=() bool Public		base_uint& [in] a base_uint& [in] b
operator!=() bool		base_uint& [in] a

Method	Notes	Parameters
Public		uint64_t [in] b
Const operator&() base_uint Public		base_uint& [in] a base_uint& [in] b
operator&=() base_uint& Public		base_uint& [in] b
Const operator*() base_uint Public		base_uint& [in] a base_uint& [in] b
Const operator*() base_uint Public		base_uint& [in] a uint32_t [in] b
operator*=(()) base_uint& Public		uint32_t [in] b32
operator*=(()) base_uint& Public		base_uint& [in] b
Const operator/() base_uint Public		base_uint& [in] a base_uint& [in] b
operator/=(()) base_uint& Public		base_uint& [in] b
Const operator^() base_uint Public		base_uint& [in] a base_uint& [in] b
operator^=(()) base_uint& Public		base_uint& [in] b
operator^=(()) base_uint& Public		uint64_t [in] b
Const operator () base_uint Public		base_uint& [in] a base_uint& [in] b
operator =(()) base_uint& Public		base_uint& [in] b

Method	Notes	Parameters
operator =() base_uint& Public		uint64_t [in] b
Const operator~() base_uint Public		
Const operator+(()) base_uint Public		base_uint& [in] a base_uint& [in] b
operator++() base_uint& Public		
Const operator++() base_uint Public		int [in] prm1
operator+=(()) base_uint& Public		base_uint& [in] b
operator+=(()) base_uint& Public		uint64_t [in] b64
operator<() bool Public		base_uint& [in] a base_uint& [in] b
Const operator<<() base_uint Public		base_uint& [in] a int [in] shift
operator<<=() base_uint& Public		unsigned int [in] shift
operator<=() bool Public		base_uint& [in] a base_uint& [in] b
operator=() base_uint& Public		base_uint& [in] b
operator=() base_uint& Public		uint64_t [in] b
operator==() bool Public		base_uint& [in] a base_uint& [in] b
operator==() bool Public		base_uint& [in] a uint64_t [in] b

Method	Notes	Parameters
operator>() bool Public		base_uint& [in] a base_uint& [in] b
operator>=() bool Public		base_uint& [in] a base_uint& [in] b
Const operator>>() base_uint Public		base_uint& [in] a int [in] shift
operator>>=() base_uint& Public		unsigned int [in] shift
SetHex() void Public		char* [in] psz
SetHex() void Public		std::string& [in] str
size() unsigned int Public		
ToString() std::string Public		

<anonymous>

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {894FF289-2D86-4385-9CDA-7FB5CD5073E1}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
WIDTH Protected		<i>Default:</i> BITS/32

bignum_error

Type: **Class** **std::runtime_error**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {E6C8A0CA-6286-43a9-9B70-16B1179D2A47}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
bignum_error() Public		std::string& [in] str

bloomflags

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {CA2C96FB-8770-4d72-AFAB-71F8987F10A9}

First two bits of nFlags control how much IsRelevantAndUpdate actually updates The remaining bits are reserved

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
BLOOM_UPDATE_NO NE Public		<i>Default:</i> 0
BLOOM_UPDATE_ALL Public		<i>Default:</i> 1
BLOOM_UPDATE_P2P UBKEY_ONLY Public	Only adds outpoints to the filter if the output is a pay-to-pubkey/pay-to-multisig script	<i>Default:</i> 2
BLOOM_UPDATE_MA SK Public		<i>Default:</i> 3

ecgroup_order

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {4028A109-7D15-4940-B850-A3A6E52FA771}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
pgroup EC_GROUP*	Private	<i>Default:</i>

Operations

Method	Notes	Parameters
ecgroup_order() Private		
~ecgroup_order() Private		
Static Const get() EC_GROUP* Public		

int16_t

Type: Class signed short
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {45A2050F-F152-4cb0-A37A-C2D71CE09D0E}

Custom Properties

- isActive = False

int32_t

Type: Class signed int
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A6E12C77-0E14-4d46-ABBA-367C7E456E41}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTransaction	Public nVersion int32_t	
Association Source -> Destination	Public CBlockHeader	Public nVersion int32_t	
Association Source -> Destination	Public CMutableTransaction	Public nVersion int32_t	
Association Source -> Destination	Public CTransaction	Public CURRENT_VERSION int32_t	
Association Source -> Destination	Public CBlockHeader	Public CURRENT_VERSION int32_t	

int64_t

Type: **Class** **signed long long**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {2CD16D6E-D72C-48dd-BE73-589CDDE946EC}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public RecentRequestEntry	Public id int64_t	
Association Source -> Destination	Public CScriptNum	Private m_value int64_t	
Association Source -> Destination	Public CompactionStats	Public bytes_read int64_t	
Association Source -> Destination	Public CompactionStats	Public bytes_written int64_t	
Association	Public	Public micros	

Connector	Source	Target	Notes
Source -> Destination	CompactionStats	int64_t	
Association Source -> Destination	Public Params	Public nPowTargetSpacing int64_t	
Association Source -> Destination	Public Params	Public nPowTargetTimespan int64_t	
Association Source -> Destination	Public Compaction	Private overlapped_bytes_ int64_t	
Association Source -> Destination	Public CBanEntry	Public nCreateTime int64_t	
Association Source -> Destination	Public CNodeStats	Public nLastRecv int64_t	
Association Source -> Destination	Public CNodeStats	Public nLastSend int64_t	
Association Source -> Destination	Public CNodeStats	Public nTimeOffset int64_t	
Association Source -> Destination	Public CNode	Public nLastRecv int64_t	
Association Source -> Destination	Public CNode	Public nLastSend int64_t	
Association Source -> Destination	Public CNode	Public nPingUsecStart int64_t	
Association Source -> Destination	Public CNode	Public nPingUsecTime int64_t	
Association Source -> Destination	Public CNodeStats	Public nTimeConnected int64_t	
Association Source -> Destination	Public CCheckpointData	Public nTimeLastCheckpoint int64_t	
Association Source -> Destination	Public CTxMemPoolEntry	Private nTime int64_t	
Association Source -> Destination	Public CNode	Public nTimeOffset int64_t	
Association Source -> Destination	Public CNetMessage	Public nTime int64_t	

Connector	Source	Target	Notes
Association Source -> Destination	Public CBanEntry	Public nBanUntil int64_t	
Association Source -> Destination	Public CNode	Public nTimeConnected int64_t	
Association Source -> Destination	Public CAccountingEntry	Public nOrderPos int64_t	
Association Source -> Destination	Public CAccountingEntry	Public nTime int64_t	
Association Source -> Destination	Public CAddrInfo	Private nLastSuccess int64_t	
Association Source -> Destination	Public CAddrInfo	Public nLastTry int64_t	
Association Source -> Destination	Public CUncsignedAlert	Public nExpiration int64_t	
Association Source -> Destination	Public CUncsignedAlert	Public nRelayUntil int64_t	
Generalization Source -> Destination	Public CAmount	Public int64_t	
Association Source -> Destination	Public CKeyMetadata	Public nCreateTime int64_t	
Association Source -> Destination	Public CKeyPool	Public nTime int64_t	
Association Source -> Destination	Public CReserveKey	Protected nIndex int64_t	
Association Source -> Destination	Public CWalletTx	Public nOrderPos int64_t	
Association Source -> Destination	Public CWallet	Private nLastResend int64_t	
Association Source -> Destination	Public CWallet	Private nNextResend int64_t	
Association Source -> Destination	Public CWallet	Public nOrderPosNext int64_t	
Association Source -> Destination	Public CWallet	Public nTimeFirstKey int64_t	
Association Source -> Destination	Public CWalletKey	Public nTimeCreated int64_t	

Connector	Source	Target	Notes
Association Source -> Destination	Public CWalletKey	Public nTimeExpires int64_t	
Association Source -> Destination	Public CCheckpointData	Public nTransactionsLastChec kpoint int64_t	

int8_t

Type: **Class** **signed char**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {6BBC8220-284B-46bb-9B3C-2D9520527D3C}

Define C99 equivalent types.

Custom Properties

- isActive = False

isminefilter

Type: **Class** **uint8_t**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {9D457D73-AA4D-4919-8E87-F502B065747C}

used for bitflags of isminetype

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public isminefilter	Public uint8_t	

isminetype

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {DB64F3ED-841D-4a9c-86D4-ED71EE5A42E0}

IsMine() return codes

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
ISMINE_NO Public		<i>Default:</i> 0
ISMINE_WATCH_UNSOLVABLE Public	! Indicates that we dont know how to create a scriptSig that would solve this if we were given the appropriate private keys	<i>Default:</i> 1
ISMINE_WATCH_SOLVABLE Public	! Indicates that we know how to create a scriptSig that would solve this if we were given the appropriate private keys	<i>Default:</i> 2
ISMINE_WATCH_ONLY Public		<i>Default:</i> ISMINE_WATCH_SOLVABLE ISMINE_WATCH_UNSOLVABLE

Attribute	Notes	Constraints and tags
ISMINE_SPENDABLE Public		<i>Default:</i> 4
ISMINE_ALL Public		<i>Default:</i> ISMINE_WATCH_ONLY ISMINE_SPENDABLE

jtokentype

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {39FA684B-E2C9-4c1d-90A1-00868D5B0AB1}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
JTOK_ERR Public		<i>Default:</i> -1
JTOK_NONE Public	eof	<i>Default:</i> 0

Attribute	Notes	Constraints and tags
JTOK_OBJ_OPEN Public		<i>Default:</i>
JTOK_OBJ_CLOSE Public		<i>Default:</i>
JTOK_ARR_OPEN Public		<i>Default:</i>
JTOK_ARR_CLOSE Public		<i>Default:</i>
JTOK_COLON Public		<i>Default:</i>
JTOK_COMMA Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
JTOK_KW_NULL Public		<i>Default:</i>
JTOK_KW_TRUE Public		<i>Default:</i>
JTOK_KW_FALSE Public		<i>Default:</i>
JTOK_NUMBER Public		<i>Default:</i>
JTOK_STRING Public		<i>Default:</i>

leveldb_error

Type: **Class** **std::runtime_error**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 GUID: {D9E37307-03ED-4faa-AB82-9D27FCC05AEF}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
leveldb_error() Public		std::string& [in] msg

limitedmap**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {544FE3C8-D9A1-4b47-A6DF-1A241794BA3D}

STL-like map container that only keeps the N elements with the highest value.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public limitedmap	Protected nMaxSize size_type	

Attributes

Attribute	Notes	Constraints and tags
map std::map<K, V> Protected		<i>Default:</i>

Attribute	Notes	Constraints and tags
nMaxSize size_type Protected		<i>Default:</i>
rmap std::multimap<V, iterator> Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
begin() const_iterator Public		
count() size_type Public		key_type& [in] k
empty() bool Public		
end() const_iterator Public		
erase() void Public		key_type& [in] k
find() const_iterator Public		key_type& [in] k
insert() void Public		value_type& [in] x
limitedmap() Public		size_type [in] nMaxSizeIn
max_size() size_type Public		
max_size() size_type Public		size_type [in] s
size() size_type Public		
update() void Public		const_iterator [in] itIn mapped_type& [in] v

const_iterator

Type: **Class** [std::map<K, V>::const iterator](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {BC9A7BAE-5329-4921-AF40-65DDD2243331}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

iterator

Type: **Class** [std::map<K, V>::iterator](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {8D600D16-6F6F-427b-8B74-72DF5FF42DF7}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

key_type

Type: **Class** [K](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {9144FF6C-3BED-4065-BE8F-B6200B886CEC}

Custom Properties

- isActive = False

mapped_type

Type: **Class** **V**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model Keywords:
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {5295DE13-B3AC-4f10-ACCA-98256D6D0437}

Custom Properties

- isActive = False

rmap_iterator

Type: **Class** **std::multimap<V, iterator>::iterator**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model Keywords:
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {34BEF77C-050F-41c4-B360-F5F44AA70EEF}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

size_type

Type: **Class** **std::map<K, V>::size_type**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model Keywords:
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {12C63C87-D709-4f27-A2B2-FFB58027E58F}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public limitedmap	Protected nMaxSize size_type	

value_type

Type: **Class** **std::pair<const key_type, mapped_type>**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 GUID: {4EF4D31C-41F0-45ea-927A-65A88FD81F31}

Custom Properties

- isActive = False

mapValue_t

Type: **Class** **std::map<std::string, std::string>**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 GUID: {8190AC36-4BC5-4530-8E6F-086693D491F8}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CAccountingEntry	Public mapValue mapValue_t	
Association Source -> Destination	Public CWalletTx	Public mapValue mapValue_t	

mruset

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {DA1B1A1B-93A2-493b-BA9E-79CD55EA2353}

STL-like set container that only keeps the most recent N elements.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public mruset	Protected first_unused size_type	
Association Source -> Destination	Public mruset	Protected first_used size_type	
Association Source -> Destination	Public mruset	Protected nMaxSize size_type	

Attributes

Attribute	Notes	Constraints and tags
first_unused size_type Protected		<i>Default:</i>

Attribute	Notes	Constraints and tags
first_used size_type Protected		<i>Default:</i>
nMaxSize size_type Protected Const		<i>Default:</i>
order std::vector<iterator> Protected		<i>Default:</i>
set std::set<T> Protected		<i>Default:</i>

Operations

Method	Notes	Parameters
begin() iterator Public		
clear() void Public		
count() size_type Public		key_type& [in] k
empty() bool Public		
end() iterator Public		
find() iterator Public		key_type& [in] k
insert() std::pair<iterator, bool> Public		key_type& [in] x

Method	Notes	Parameters
max_size() size_type Public		
mruset() Public		size_type [in] nMaxSizeIn
operator<() bool Public		mruset<T>& [in] a mruset<T>& [in] b
operator==() bool Public		mruset<T>& [in] a mruset<T>& [in] b
operator==() bool Public		mruset<T>& [in] a std::set<T>& [in] b
size() size_type Public		

const_iterator

Type: [Class std::set<T>::const_iterator](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {7FA0384F-F775-4d7a-988E-DFD2329760E5}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

iterator

Type: [Class std::set<T>::iterator](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {3A34E61A-CA1A-4613-B930-5023A26130B6}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

key_type

Type: **Class** **T**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 GUID: {D67C2E96-5693-43ad-AC22-D0ECC05D48CC}

Custom Properties

- isActive = False

size_type

Type: **Class** **std::set<T>::size_type**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 GUID: {04A1D12D-0915-4aab-980D-A277E835ED59}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

Connections

Connector	Source	Target	Notes
-----------	--------	--------	-------

Connector	Source	Target	Notes
Association Source -> Destination	Public mruset	Protected first_unused size_type	
Association Source -> Destination	Public mruset	Protected first_used size_type	
Association Source -> Destination	Public mruset	Protected nMaxSize size_type	

value_type

Type: [Class](#) [T](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords](#):
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {4D8B75F4-C1FD-4e97-A4AA-8EC3C6F5C015}

Custom Properties

- isActive = False

opcodetype

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords](#):
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {E17E976E-0358-4397-A1F5-D09D5EC8128D}

Script opcodes

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
OP_0 Public	push value	<i>Default:</i> 0x00
OP_FALSE Public		<i>Default:</i> OP_0
OP_PUSHDATA1 Public		<i>Default:</i> 0x4c
OP_PUSHDATA2 Public		<i>Default:</i> 0x4d
OP_PUSHDATA4 Public		<i>Default:</i> 0x4e
OP_1NEGATE Public		<i>Default:</i> 0x4f

Attribute	Notes	Constraints and tags
OP_RESERVED Public		<i>Default:</i> 0x50
OP_1 Public		<i>Default:</i> 0x51
OP_TRUE Public		<i>Default:</i> OP_1
OP_2 Public		<i>Default:</i> 0x52
OP_3 Public		<i>Default:</i> 0x53
OP_4 Public		<i>Default:</i> 0x54

Attribute	Notes	Constraints and tags
OP_5 Public		<i>Default:</i> 0x55
OP_6 Public		<i>Default:</i> 0x56
OP_7 Public		<i>Default:</i> 0x57
OP_8 Public		<i>Default:</i> 0x58
OP_9 Public		<i>Default:</i> 0x59
OP_10 Public		<i>Default:</i> 0x5a

Attribute	Notes	Constraints and tags
OP_11 Public		<i>Default:</i> 0x5b
OP_12 Public		<i>Default:</i> 0x5c
OP_13 Public		<i>Default:</i> 0x5d
OP_14 Public		<i>Default:</i> 0x5e
OP_15 Public		<i>Default:</i> 0x5f
OP_16 Public		<i>Default:</i> 0x60

Attribute	Notes	Constraints and tags
OP_NOP Public	control	<i>Default:</i> 0x61
OP_VER Public		<i>Default:</i> 0x62
OP_IF Public		<i>Default:</i> 0x63
OP_NOTIF Public		<i>Default:</i> 0x64
OP_VERIF Public		<i>Default:</i> 0x65
OP_VERNOTIF Public		<i>Default:</i> 0x66

Attribute	Notes	Constraints and tags
OP_ELSE Public		<i>Default:</i> 0x67
OP_ENDIF Public		<i>Default:</i> 0x68
OP_VERIFY Public		<i>Default:</i> 0x69
OP_RETURN Public		<i>Default:</i> 0x6a
OP_TOALTSTACK Public	stack ops	<i>Default:</i> 0x6b
OP_FROMALTSTACK Public		<i>Default:</i> 0x6c

Attribute	Notes	Constraints and tags
OP_2DROP Public		<i>Default:</i> 0x6d
OP_2DUP Public		<i>Default:</i> 0x6e
OP_3DUP Public		<i>Default:</i> 0x6f
OP_2OVER Public		<i>Default:</i> 0x70
OP_2ROT Public		<i>Default:</i> 0x71
OP_2SWAP Public		<i>Default:</i> 0x72

Attribute	Notes	Constraints and tags
OP_IFDUP Public		<i>Default:</i> 0x73
OP_DEPTH Public		<i>Default:</i> 0x74
OP_DROP Public		<i>Default:</i> 0x75
OP_DUP Public		<i>Default:</i> 0x76
OP_NIP Public		<i>Default:</i> 0x77
OP_OVER Public		<i>Default:</i> 0x78

Attribute	Notes	Constraints and tags
OP_PICK Public		<i>Default:</i> 0x79
OP_ROLL Public		<i>Default:</i> 0x7a
OP_ROT Public		<i>Default:</i> 0x7b
OP_SWAP Public		<i>Default:</i> 0x7c
OP_TUCK Public		<i>Default:</i> 0x7d
OP_CAT Public	splice ops	<i>Default:</i> 0x7e

Attribute	Notes	Constraints and tags
OP_SUBSTR Public		<i>Default:</i> 0x7f
OP_LEFT Public		<i>Default:</i> 0x80
OP_RIGHT Public		<i>Default:</i> 0x81
OP_SIZE Public		<i>Default:</i> 0x82
OP_INVERT Public	bit logic	<i>Default:</i> 0x83
OP_AND Public		<i>Default:</i> 0x84

Attribute	Notes	Constraints and tags
OP_OR Public		<i>Default:</i> 0x85
OP_XOR Public		<i>Default:</i> 0x86
OP_EQUAL Public		<i>Default:</i> 0x87
OP_EQUALVERIFY Public		<i>Default:</i> 0x88
OP_RESERVED1 Public		<i>Default:</i> 0x89
OP_RESERVED2 Public		<i>Default:</i> 0x8a

Attribute	Notes	Constraints and tags
OP_1ADD Public	numeric	<i>Default:</i> 0x8b
OP_1SUB Public		<i>Default:</i> 0x8c
OP_2MUL Public		<i>Default:</i> 0x8d
OP_2DIV Public		<i>Default:</i> 0x8e
OP_NEGATE Public		<i>Default:</i> 0x8f
OP_ABS Public		<i>Default:</i> 0x90

Attribute	Notes	Constraints and tags
OP_NOT Public		<i>Default:</i> 0x91
OP_0NOTEQUAL Public		<i>Default:</i> 0x92
OP_ADD Public		<i>Default:</i> 0x93
OP_SUB Public		<i>Default:</i> 0x94
OP_MUL Public		<i>Default:</i> 0x95
OP_DIV Public		<i>Default:</i> 0x96

Attribute	Notes	Constraints and tags
OP_MOD Public		<i>Default:</i> 0x97
OP_LSHIFT Public		<i>Default:</i> 0x98
OP_RSHIFT Public		<i>Default:</i> 0x99
OP_BOOLAND Public		<i>Default:</i> 0x9a
OP_BOOLOR Public		<i>Default:</i> 0x9b
OP_NUMEQUAL Public		<i>Default:</i> 0x9c

Attribute	Notes	Constraints and tags
OP_NUMEQUALVERIFY Y Public		<i>Default:</i> 0x9d
OP_NUMNOTEQUAL Public		<i>Default:</i> 0x9e
OP_LESSTHAN Public		<i>Default:</i> 0x9f
OP_GREATERTHAN Public		<i>Default:</i> 0xa0
OP_LESSTHANOREQUAL Public		<i>Default:</i> 0xa1
OP_GREATERTHANOREQUAL Public		<i>Default:</i> 0xa2

Attribute	Notes	Constraints and tags
OP_MIN Public		<i>Default:</i> 0xa3
OP_MAX Public		<i>Default:</i> 0xa4
OP_WITHIN Public		<i>Default:</i> 0xa5
OP_RIPEMD160 Public	crypto	<i>Default:</i> 0xa6
OP_SHA1 Public		<i>Default:</i> 0xa7
OP_SHA256 Public		<i>Default:</i> 0xa8

Attribute	Notes	Constraints and tags
OP_HASH160 Public		<i>Default:</i> 0xa9
OP_HASH256 Public		<i>Default:</i> 0xaa
OP_CODESEPARATOR Public		<i>Default:</i> 0xab
OP_CHECKSIG Public		<i>Default:</i> 0xac
OP_CHECKSIGVERIFY Y Public		<i>Default:</i> 0xad
OP_CHECKMULTISIG Public		<i>Default:</i> 0xae

Attribute	Notes	Constraints and tags
OP_CHECKMULTISIG VERIFY Public		<i>Default:</i> 0xaf
OP_NOP1 Public	expansion	<i>Default:</i> 0xb0
OP_NOP2 Public		<i>Default:</i> 0xb1
OP_CHECKLOCKTIMEVERIFY Public		<i>Default:</i> OP_NOP2
OP_NOP3 Public		<i>Default:</i> 0xb2
OP_NOP4 Public		<i>Default:</i> 0xb3

Attribute	Notes	Constraints and tags
OP_NOP5 Public		<i>Default:</i> 0xb4
OP_NOP6 Public		<i>Default:</i> 0xb5
OP_NOP7 Public		<i>Default:</i> 0xb6
OP_NOP8 Public		<i>Default:</i> 0xb7
OP_NOP9 Public		<i>Default:</i> 0xb8
OP_NOP10 Public		<i>Default:</i> 0xb9

Attribute	Notes	Constraints and tags
OP_SMALLDATA Public	template matching params	<i>Default:</i> 0xf9
OP_SMALLINTEGER Public		<i>Default:</i> 0xfa
OP_PUBKEYS Public		<i>Default:</i> 0xfb
OP_PUBKEYHASH Public		<i>Default:</i> 0xfd
OP_PUBKEY Public		<i>Default:</i> 0xfe
OP_INVALIDOPCODE Public		<i>Default:</i> 0xff

proxyType

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {45B75443-8D17-4cf8-9CAA-516CAC9FA009}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public proxyType	Public proxy CService	

Attributes

Attribute	Notes	Constraints and tags
proxy CService Public		<i>Default:</i>
randomize_credentials bool Public		<i>Default:</i>

Operations

Method	Notes	Parameters
IsValid() bool Public		
proxyType() Public		
proxyType() Public		CService& [in] proxy bool [in] randomize_credentials

scriptnum_error

Type: Class **std::runtime_error**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {050EC08B-266F-41eb-BA17-7F4D2C413DB6}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
scriptnum_error() Public		std::string& [in] str

secp256k1_ecdsa_sig_t

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D40302D0-319D-433e-B0A8-BB4AFA8BE7A6}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
<u>Association</u> Source -> Destination	Public secp256k1_ecdsa_sig_t	Public r secp256k1_scalar_t	
<u>Association</u> Source -> Destination	Public secp256k1_ecdsa_sig_t	Public s secp256k1_scalar_t	

Attributes

Attribute	Notes	Constraints and tags
r secp256k1_scalar_t Public		<i>Default:</i>
s secp256k1_scalar_t Public		<i>Default:</i>

secp256k1_ecmult_context_t

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {DA977C48-6739-418c-B8A2-5F864E73FD3D}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
secp256k1_ge_storage_t int Public Collection	odd multiples of $2^{128} * \text{generator}$	<i>Default:</i>
secp256k1_ge_storage_t int Public Collection	odd multiples of the generator	<i>Default:</i>

secp256k1_ecmult_gen_context_t

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {AFC4B86A-56B7-45ca-8F92-F313D466422C}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_ecmult_gen_context_t	Public initial secp256k1_gej_t	
Association Source -> Destination	Public secp256k1_ecmult_gen_context_t	Public blind secp256k1_scalar_t	

Attributes

Attribute	Notes	Constraints and tags
blind secp256k1_scalar_t Public		<i>Default:</i>
initial secp256k1_gej_t Public		<i>Default:</i>
secp256k1_ge_storage_t int Public Collection	prec[j][i] = $16^j * i * G + U_i$	<i>Default:</i>

secp256k1_fe_storage_t

Type: [**Class**](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {BEE9E59C-B27D-42e1-8FB8-2F5BB76D5DB4}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_fe_storage_t	Public n uint64_t	

Attributes

Attribute	Notes	Constraints and tags
n uint64_t Public Collection		<i>Default:</i>

secp256k1_fe_storage_t

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {44C68318-2D46-4976-8D12-CBFE02BC643C}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_fe_storage_t	Public n uint32_t	
Association Source -> Destination	Public secp256k1_ge_storage_t	Public x secp256k1_fe_storage_t	
Association Source -> Destination	Public secp256k1_ge_storage_t	Public y secp256k1_fe_storage_t	

Attributes

Attribute	Notes	Constraints and tags
n uint32_t Public Collection		<i>Default:</i>

secp256k1_fe_t

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {27162589-5FD7-4f46-A872-E818DD82F063}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_fe_t	Public n uint64_t	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
magnitude int Public		<i>Default:</i>
n uint64_t Public Collection	X = sum(i=0..4, elem[i]*2^52) mod n	<i>Default:</i>
normalized int Public		<i>Default:</i>

secp256k1_fe_t

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model

Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {DCD17118-42EC-4413-9A9D-12C88C4D55FE}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
Association	Public	Public x	

Connector	Source	Target	Notes
Source -> Destination	secp256k1_ge_t	secp256k1_fe_t	
Association Source -> Destination	Public secp256k1_ge_t	Public y secp256k1_fe_t	
Association Source -> Destination	Public secp256k1_gej_t	Public x secp256k1_fe_t	
Association Source -> Destination	Public secp256k1_gej_t	Public y secp256k1_fe_t	
Association Source -> Destination	Public secp256k1_gej_t	Public z secp256k1_fe_t	
Association Source -> Destination	Public secp256k1_fe_t	Public n uint32_t	

Attributes

Attribute	Notes	Constraints and tags
magnitude int Public		<i>Default:</i>
n uint32_t Public Collection	X = sum(i=0..9, elem[i]*2^26) mod n	<i>Default:</i>
normalized int Public		<i>Default:</i>

secp256k1_ge_storage_t

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {C740D675-F413-46f7-A8FF-BA52A8006AD6}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_ge_storage_t	Public x secp256k1_fe_storage_t	
Association Source -> Destination	Public secp256k1_ge_storage_t	Public y secp256k1_fe_storage_t	

Attributes

Attribute	Notes	Constraints and tags
x secp256k1_fe_storage_t Public		<i>Default:</i>
y secp256k1_fe_storage_t Public		<i>Default:</i>

secp256k1_ge_t

Type:

Class

Status:

Proposed. Version 1.0. Phase 1.0.

Package: Class Model *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {7849A23C-622F-4860-94EB-8F54845B2908}

A group element of the secp256k1 curve, in affine coordinates.

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_ge_t	Public x secp256k1_fe_t	
Association Source -> Destination	Public secp256k1_ge_t	Public y secp256k1_fe_t	

Attributes

Attribute	Notes	Constraints and tags
infinity int Public	whether this represents the point at infinity	<i>Default:</i>
x secp256k1_fe_t Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
y secp256k1_fe_t Public		<i>Default:</i>

secp256k1_gej_t

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model

Keywords:

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {D87B45A5-6F76-4604-9DEF-20B5F1B680EF}

A group element of the secp256k1 curve, in jacobian coordinates.

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_gej_t	Public x secp256k1_fe_t	
Association Source -> Destination	Public secp256k1_gej_t	Public y secp256k1_fe_t	
Association Source -> Destination	Public secp256k1_gej_t	Public z secp256k1_fe_t	
Association Source -> Destination	Public secp256k1_ecmult_gen_context_t	Public initial secp256k1_gej_t	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
infinity int Public	whether this represents the point at infinity	<i>Default:</i>
x secp256k1_fe_t Public	actual X: x/z^2	<i>Default:</i>
y secp256k1_fe_t Public	actual Y: y/z^3	<i>Default:</i>
z secp256k1_fe_t Public		<i>Default:</i>

secp256k1_hmac_sha256_t

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {0791DA09-B445-41d0-BC95-DF7B12F5CE0F}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.

- `typedef = true.`

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_hmac_sha256_t	Public inner secp256k1_sha256_t	
Association Source -> Destination	Public secp256k1_hmac_sha256_t	Public outer secp256k1_sha256_t	

Attributes

Attribute	Notes	Constraints and tags
inner secp256k1_sha256_t Public		<i>Default:</i>
outer secp256k1_sha256_t Public		<i>Default:</i>

secp256k1_num_t**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {FB0BA4A7-6051-4f2a-A86C-1EBD4BA9DB01}***Custom Properties***

- `isActive = False`

Tagged Values

- anonymous = true.
- typedef = true.

Attributes

Attribute	Notes	Constraints and tags
data mp_limb_t Public Collection		<i>Default:</i>
limbs int Public		<i>Default:</i>
neg int Public		<i>Default:</i>

secp256k1_rfc6979_hmac_sha256_t

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {DE86EADA-492A-4e50-95C7-4BB1A7C2C3F9}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.

- `typedef = true.`

Attributes

Attribute	Notes	Constraints and tags
k unsigned char Public Collection		<i>Default:</i>
retry int Public		<i>Default:</i>
v unsigned char Public Collection		<i>Default:</i>

secp256k1_scalar_t

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {99EF4747-3B8C-415e-B6C0-78F2ABA916B2}

A scalar modulo the group order of the secp256k1 curve.

Custom Properties

- `isActive = False`

Tagged Values

- `anonymous = true.`
- `typedef = true.`

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_scalar_t	Public d uint32_t	

Attributes

Attribute	Notes	Constraints and tags
d uint32_t Public Collection		<i>Default:</i>

secp256k1_scalar_t

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {C716EE63-7A4A-4bbc-8607-5C38FE25A54B}

A scalar modulo the group order of the secp256k1 curve.

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_ecmult_gen _context_t	Public blind secp256k1_scalar_t	
Association Source -> Destination	Public secp256k1_scalar_t	Public d uint64_t	

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_ecdsa_sig_t	Public r secp256k1_scalar_t	
Association Source -> Destination	Public secp256k1_ecdsa_sig_t	Public s secp256k1_scalar_t	

Attributes

Attribute	Notes	Constraints and tags
d uint64_t Public Collection		<i>Default:</i>

secp256k1_sha256_t

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {194EDAE6-F9A2-4c6d-BB64-5B5E36FE518B}

Custom Properties

- isActive = False

Tagged Values

- anonymous = true.
- typedef = true.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_hmac_sha256_t	Public inner secp256k1_sha256_t	
Association Source -> Destination	Public secp256k1_sha256_t	Public s uint32_t	

Connector	Source	Target	Notes
Association Source -> Destination	Public secp256k1_hmac_sha256_t	Public outer secp256k1_sha256_t	
Association Source -> Destination	Public secp256k1_sha256_t	Public buf uint32_t	

Attributes

Attribute	Notes	Constraints and tags
buf uint32_t Public Collection	In big endian	<i>Default:</i>
bytes size_t Public		<i>Default:</i>
s uint32_t Public Collection		<i>Default:</i>

secure_allocator

Type: **Class** **std::allocator**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {89818C8A-C1EB-4f6a-A5DC-6F701EBF204F}

Allocator that locks its contents from being paged out of memory and clears its contents before deletion.

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
allocate() T*		std::size_t [in] n void* [in] hint
Public		
deallocate() void		T* [in] p std::size_t [in] n
Public		
secure_allocator()		
Public		
secure_allocator()		secure_allocator& [in] a
Public		
secure_allocator()		secure_allocator<U>& [in] a
Public		
~secure_allocator()		
Public		

base

Type: Class **std::allocator<T>**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {F7AFB476-373E-4f97-8B76-FED3CDF50A1F}

MSVC8 default copy constructor is broken

Custom Properties

- isActive = False

const_pointer

Type: Class **base::const_pointer**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {CB181927-C47D-4558-9E1D-C29EF72DDE47}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

const_reference

Type: **Class** base::const_reference
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model Keywords:
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {250E5455-D90F-440e-87C1-C1CA4F4AC8AC}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

difference_type

Type: **Class** base::difference_type
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model Keywords:
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {0909EDD5-0AB4-49cd-BE08-96DCAD93AA66}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

pointer

Type: **Class** **base::pointer**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {2A6920B9-0719-4a17-B25B-A2055767B1F5}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

rebind

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {8A273CBA-E379-423f-AAAF-7F59B31CADD}

Custom Properties

- isActive = False

other

Type: **Class** **secure_allocator< Other>**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {F32E50F2-3C79-42c8-81DA-E81C3991A60A}

Custom Properties

- isActive = False

reference

Type: **Class** **base::reference**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {8D7395FF-D80E-47e6-85EF-15A3946B28ED}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

size_type

Type: **Class** **base::size_type**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {185A3B75-D8B2-43ff-AEEF-96DA4E518998}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

value_type

Type: **Class** **base::value_type**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {7E70AC1F-DBF1-4657-B2FD-92E7ED956903}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

txnouttype**Type:** Enumeration**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** Class Model **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {B37CC39D-811F-4ffe-BB4F-820D50646B0F}**Custom Properties**

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
TX_NONSTANDARD Public		<i>Default:</i>
TX_PUBKEY Public	'standard' transaction types:	<i>Default:</i>
TX_PUBKEYHASH Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
TX_SCRIPTHASH Public		<i>Default:</i>
TX_MULTISIG Public		<i>Default:</i>
TX_NULL_DATA Public		<i>Default:</i>

uint160

Type: [Class](#) [base blob](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model [Keywords](#):
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {9A7C087E-89B4-4697-9D9F-EBF08B18BA25}

160-bit opaque blob.

@note This type is called uint160 for historical reasons only. It is an opaque blob of 160 bits and has no integer operations.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public CScriptID	Public uint160	
Generalization Source -> Destination	Public CKeyID	Public uint160	

Connector	Source	Target	Notes
Generalization Source -> Destination	Public uint160	Public base_blob	

Operations

Method	Notes	Parameters
uint160() Public		
uint160() Public		base_blob<160>& [in] b
uint160() Public		std::vector<unsigned char>& [in] vch

uint16_t

Type: **Class** **unsigned short**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {58345EB4-7FD1-44ed-823D-FA5FDA285565}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public SeedSpec6	Public port uint16_t	

uint256

Type: **Class** **base blob**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {10E6C106-150C-4862-88A5-865593A60FDC}

256-bit opaque blob.

@note This type is called uint256 for historical reasons only. It is an opaque blob of 256 bits and has no integer operations. Use arith_uint256 if those are required.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
<u>Association</u> Source -> Destination	Public CBlockHeader	Public hashMerkleRoot uint256	
<u>Association</u> Source -> Destination	Public CBlockHeader	Public hashPrevBlock uint256	
<u>Association</u> Source -> Destination	Public CTransaction	Private hash uint256	
<u>Association</u> Source -> Destination	Public COutPoint	Public hash uint256	
<u>Association</u> Source -> Destination	Public Params	Public hashGenesisBlock uint256	
<u>Association</u> Source -> Destination	Public TransactionRecord	Public hash uint256	
<u>Association</u> Source -> Destination	Public Params	Public powLimit uint256	
<u>Association</u> Source -> Destination	Public CNode	Public hashContinue uint256	
<u>Association</u> Source -> Destination	Public CDiskBlockIndex	Public hashPrev uint256	
<u>Association</u> Source -> Destination	Public CInv	Public hash uint256	
<u>Generalization</u> Source -> Destination	Public uint256	Public base_blob	
<u>Generalization</u> Source -> Destination	Public ChainCode	Public uint256	
<u>Generalization</u> Source -> Destination	Public ChainCode	Public uint256	
<u>Association</u> Source -> Destination	Public CAddrMan	Private nKey uint256	
<u>Association</u> Source -> Destination	Public CMerkleTx	Public hashBlock uint256	

Connector	Source	Target	Notes
Association Source -> Destination	Public CCoinsKeyHasher	Private salt uint256	
Association Source -> Destination	Public CCoinsStats	Public hashBlock uint256	
Association Source -> Destination	Public CCoinsViewCache	Protected hashBlock uint256	
Association Source -> Destination	Public CCoinsStats	Public hashSerialized uint256	
Association Source -> Destination	Public CBlockIndex	Public hashMerkleRoot uint256	
Association Source -> Destination	Public CBlockIndex	Public phashBlock uint256	

Operations

Method	Notes	Parameters
GetCheapHash() uint64_t Public		
GetHash() uint64_t Public		uint256& [in] salt
uint256() Public		
uint256() Public		base_blob<256>& [in] b
uint256() Public		std::vector<unsigned char>& [in] vch

uint32_t

Type: **Class** **unsigned int**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {EE67C200-176A-4f49-A1B0-B92DB888F66A}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CTransaction	Public nLockTime uint32_t	
Association Source -> Destination	Public CBlockHeader	Public nBits uint32_t	
Association Source -> Destination	Public CBlockHeader	Public nNonce uint32_t	
Association Source -> Destination	Public CBlockHeader	Public nTime uint32_t	
Association Source -> Destination	Public CMutableTransaction	Public nLockTime uint32_t	
Association Source -> Destination	Public Random	Private seed_ uint32_t	
Association Source -> Destination	Public COutPoint	Public n uint32_t	
Association Source -> Destination	Public CTxIn	Public nSequence uint32_t	
Association Source -> Destination	Public secp256k1_fe_storage_t	Public n uint32_t	
Association Source -> Destination	Public secp256k1_sha256_t	Public s uint32_t	
Association Source -> Destination	Public secp256k1_scalar_t	Public d uint32_t	
Association Source -> Destination	Public secp256k1_sha256_t	Public buf uint32_t	
Association Source -> Destination	Public secp256k1_fe_t	Public n uint32_t	
Association Source -> Destination	Public Block	Private restart_offset_ uint32_t	
Association Source -> Destination	Public Writer	Private type_crc_ uint32_t	
Association Source -> Destination	Public CRIPEMD160	Private s uint32_t	
Association Source -> Destination	Public CSHA1	Private s uint32_t	
Association	Public	Private s	

Connector	Source	Target	Notes
Source -> Destination	CSHA256	uint32_t	
Association Source -> Destination	Public DBImpl	Private seed_ uint32_t	
Association Source -> Destination	Public CInPoint	Public n uint32_t	
Association Source -> Destination	Public base_uint	Protected pn uint32_t	
Association Source -> Destination	Public CBlockIndex	Public nSequenceId uint32_t	

uint64_t

Type: **Class** **unsigned long long**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {58B59F59-9A51-4270-96AC-5C63510807CA}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public BlockHandle	Private offset_ uint64_t	
Association Source -> Destination	Public BlockHandle	Private size_ uint64_t	
Association Source -> Destination	Public secp256k1_fe_t	Public n uint64_t	
Association Source -> Destination	Public secp256k1_fe_storage_t	Public n uint64_t	
Association Source -> Destination	Public secp256k1_scalar_t	Public d uint64_t	
Association Source -> Destination	Public Reader	Private end_of_buffer_offset_ uint64_t	

Connector	Source	Target	Notes
Association Source -> Destination	Public Reader	Private last_record_offset_ uint64_t	
Association Source -> Destination	Public Reader	Private initial_offset_ uint64_t	
Association Source -> Destination	Public CSHA512	Private s uint64_t	
Association Source -> Destination	Public DBImpl	Private logfile_number_ uint64_t	
Association Source -> Destination	Public VersionSet	Private prev_log_number_ uint64_t	
Association Source -> Destination	Public VersionSet	Private log_number_ uint64_t	
Association Source -> Destination	Public Compaction	Private max_output_file_size_ uint64_t	
Association Source -> Destination	Public VersionEdit	Private prev_log_number_ uint64_t	
Association Source -> Destination	Public FileMetaData	Public file_size uint64_t	
Association Source -> Destination	Public FileMetaData	Public number uint64_t	
Association Source -> Destination	Public VersionSet	Private next_file_number_ uint64_t	
Association Source -> Destination	Public VersionEdit	Private next_file_number_ uint64_t	
Association Source -> Destination	Public VersionSet	Private manifest_file_number_ uint64_t	
Association Source -> Destination	Public VersionSet	Private last_sequence_ uint64_t	
Association Source -> Destination	Public VersionEdit	Private log_number_ uint64_t	
Association Source -> Destination	Public CNodeStats	Public nRecvBytes uint64_t	

Connector	Source	Target	Notes
Association Source -> Destination	Public CNodeStats	Public nSendBytes uint64_t	
Association Source -> Destination	Public CNodeStats	Public nServices uint64_t	
Association Source -> Destination	Public CNode	Public nPingNonceSent uint64_t	
Association Source -> Destination	Public CNode	Public nRecvBytes uint64_t	
Association Source -> Destination	Public CBufferedFile	Private nReadLimit uint64_t	
Association Source -> Destination	Public CBufferedFile	Private nReadPos uint64_t	
Association Source -> Destination	Public CBufferedFile	Private nRewind uint64_t	
Association Source -> Destination	Public CBufferedFile	Private nSrcPos uint64_t	
Association Source -> Destination	Public CTxMemPool	Private cachedInnerUsage uint64_t	
Association Source -> Destination	Public CTxMemPool	Private totalTxSize uint64_t	
Association Source -> Destination	Public CBlockFileInfo	Public nTimeFirst uint64_t	
Association Source -> Destination	Public CBlockFileInfo	Public nTimeLast uint64_t	
Association Source -> Destination	Public CNode	Public nSendBytes uint64_t	
Association Source -> Destination	Public CNode	Private nTotalBytesRecv uint64_t	
Association Source -> Destination	Public CNode	Private nTotalBytesSent uint64_t	
Association Source -> Destination	Public CAddress	Public nServices uint64_t	
Association Source -> Destination	Public CNode	Public nServices uint64_t	

Connector	Source	Target	Notes
Association Source -> Destination	Public CAccountingEntry	Public nEntryNo uint64_t	
Association Source -> Destination	Public CCoinsStats	Public nSerializedSize uint64_t	
Association Source -> Destination	Public CCoinsStats	Public nTransactionOutputs uint64_t	
Association Source -> Destination	Public CCoinsStats	Public nTransactions uint64_t	
Association Source -> Destination	Public CChainParams	Protected nPruneAfterHeight uint64_t	

uint8_t

Type: **Class** **unsigned char**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Class Model **Keywords:**
 Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
 G UID: {D6345372-3415-483f-8FF7-0084C3F7460C}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public SeedSpec6	Public addr uint8_t	
Association Source -> Destination	Public base_blob	Protected data uint8_t	
Association Source -> Destination	Public CBanEntry	Public banReason uint8_t	
Association Source -> Destination	Public CSubNet	Protected netmask uint8_t	
Generalization Source -> Destination	Public isminefilter	Public uint8_t	

uint_error

Type: **Class** [std::runtime_error](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {B58FC4EC-090B-4b60-8445-E2D0C2768D77}

[Custom Properties](#)

- isActive = False

[Operations](#)

Method	Notes	Parameters
uint_error() Public		std::string& [in] str

valtype

Type: **Class** [vector<unsigned char>](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {F5C530D0-B0A1-48ce-BD0F-37DDDB731554}

[Custom Properties](#)

- isActive = False

valtype

Type: **Class** [vector<unsigned char>](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {913FAA97-0606-42c9-A423-1A2CD994C712}

[Custom Properties](#)

Custom Properties

- isActive = False

valtype

Type: **Class** **vector<unsigned char>**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {5B4306E8-73F2-43dc-8052-820AE43BA121}

Custom Properties

- isActive = False

valtype

Type: **Class** **vector<unsigned char>**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {FB5D33F2-F12B-4192-B17B-22CA1A3F81FC}

Custom Properties

- isActive = False

zero_after_free_allocator

Type: **Class** **std::allocator**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {149761BE-BF48-490c-B041-9CDCAD37486B}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
deallocate() void Public		T* [in] p std::size_t [in] n
zero_after_free_allocator() Public		
zero_after_free_allocator() Public		zero_after_free_allocator& [in] a
zero_after_free_allocator() Public		zero_after_free_allocator<U>& [in] a
~zero_after_free_allocator() Public		

base

Type: [Class](#) [**std::allocator<T>**](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D8255B2D-BD6D-4f50-9036-D2BED731040A}

MSVC8 default copy constructor is broken

Custom Properties

- isActive = False

const_pointer

Type: [Class](#) [**base::const_pointer**](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {2C4EB1F1-0A82-4ec3-BA14-03A6A78B33F3}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

const_reference

Type: **Class** **base::const_reference**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {3C91D5E1-72C7-4930-A8BD-73E495183A19}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

difference_type

Type: **Class** **base::difference_type**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {89626AD4-6670-42b7-B34D-A37A40F2CF03}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

pointer

Type: **Class** **base::pointer**

Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {1E2D548E-99BC-4d94-B211-1410AB9FBDBD}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

rebind

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {227789BE-E829-4951-B8E1-9DE20ADB88AA}

Custom Properties

- isActive = False

other

Type: **Class** **zero_after_free_allocator< Other>**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model *Keywords:*
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {EA49BC0E-D10E-44e2-BE35-7E4AAF1E2E7E}

Custom Properties

- isActive = False

reference

Type: **Class** **base::reference**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {E4FB3704-F4A8-4bef-81D7-FD9D56F73CA1}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

size_type

Type: **Class** **base::size_type**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {589AAB5A-ED4B-48b0-BA87-394DEB4C1B88}

Custom Properties

- isActive = False

Tagged Values

- type = typename.

value_type

Type: **Class** **base::value_type**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {FC928FF3-A521-456c-8B52-AF8C02D48481}

Custom Properties

Custom Properties

- isActive = False

Tagged Values

- type = typename.

Consensus

Type:	<u>Package</u>
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015</i>
GUID:	{3C00BD18-F1C6-441d-8391-0C24A6661F8E}

Consensus - (Logical diagram)

<u>Created By:</u>	dev on 8/30/2015
<u>Last Modified:</u>	8/30/2015
<u>Version:</u>	1.0. <u>Locked</u> : False
<u>GUID:</u>	{7D0211EE-0E35-477e-9BA9-D8A5C3990B14}

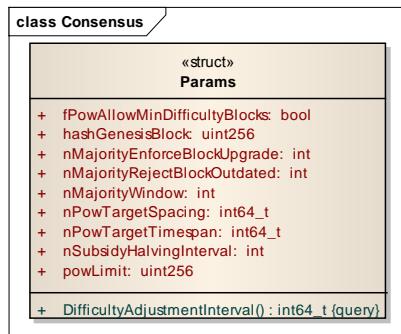


Figure: 2

Params

Type:	<u>Class</u>
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Consensus
Keywords:	
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>
GUID:	{30CB9523-9515-4f9b-8EC6-48D97DB209AC}

Parameters that influence chain consensus.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Params	Public hashGenesisBlock uint256	
Association Source -> Destination	Public Params	Public nPowTargetSpacing int64_t	
Association Source -> Destination	Public Params	Public nPowTargetTimespan int64_t	
Association Source -> Destination	Public Params	Public powLimit uint256	
Association Source -> Destination	Public CChainParams	Protected consensus Params	

Attributes

Attribute	Notes	Constraints and tags
fPowAllowMinDifficultyBlocks bool Public		<i>Default:</i>
hashGenesisBlock uint256 Public		<i>Default:</i>
nMajorityEnforceBlockUpgrade int Public	Used to check majorities for block version upgrade	<i>Default:</i>

Attribute	Notes	Constraints and tags
nMajorityRejectBlockOutdated int Public		<i>Default:</i>
nMajorityWindow int Public		<i>Default:</i>
nPowTargetSpacing int64_t Public		<i>Default:</i>
nPowTargetTimespan int64_t Public		<i>Default:</i>
nSubsidyHalvingInterval int Public		<i>Default:</i>
powLimit uint256 Public	Proof of work parameters	<i>Default:</i>

Operations

Method	Notes	Parameters
DifficultyAdjustmentInte		

Method	Notes	Parameters
rval() int64_t Public		

leveldb

Type: **Package**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Class Model

Detail: Created on 8/30/2015. Last modified on 8/30/2015

GUID: {358C403E-83C4-43c8-89C3-6890C0FC48E1}

leveldb - (Logical diagram)

Created By: dev on 8/30/2015

Last Modified: 8/30/2015

Version: 1.0. Locked: False

GUID: {14750297-84AA-4960-A7FA-FB5216201116}

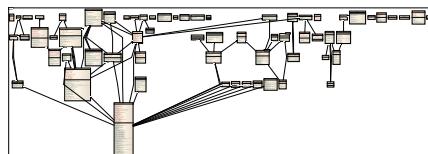


Figure: 3

Arena

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {A02E0BD6-25AA-4321-8A71-2DE0CFBD558D}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public MemTable	Private arena_ Arena	
Association Source -> Destination	Public SkipList	Private arena_ Arena	

Attributes

Attribute	Notes	Constraints and tags
alloc_bytes_remaining_ size_t Private		<i>Default:</i>
alloc_ptr_ char* Private	Allocation state	<i>Default:</i>
blocks_ std::vector<char*> Private	Array of new[] allocated memory blocks	<i>Default:</i>
blocks_memory_ size_t Private	Bytes of memory in blocks allocated so far	<i>Default:</i>

Operations

Method	Notes	Parameters
Allocate() char*		size_t [in] bytes
Public		
AllocateAligned() char*		size_t [in] bytes
Public		
AllocateFallback() char*		size_t [in] bytes
Private		
AllocateNewBlock() char*		size_t [in] block_bytes
Private		
Arena()		
Public		
~Arena()		

Method	Notes	Parameters
Public		
Arena() Private		Arena& [in] prm1
MemoryUsage() size_t Public		
operator=() void Private		Arena& [in] prm1

Block

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb *Keywords:*

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {33C442B7-069A-46ae-BF4E-8C56F225E05D}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Block	Private restart_offset_ uint32_t	

Attributes

Attribute	Notes	Constraints and tags
data_ char* Private Const		<i>Default:</i>
owned_ bool Private	Block owns data_[]	<i>Default:</i>

Attribute	Notes	Constraints and tags
restart_offset_ uint32_t Private	Offset in data_ of restart array	<i>Default:</i>
size_ size_t Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Block() Public		BlockContents& [in] contents
~Block() Public		
Block() Private		Block& [in] prm1
NewIterator() Iterator* Public		Comparator* [in] comparator
NumRestarts() uint32_t Private		
operator=() void Private		Block& [in] prm1
size() size_t Public		

BlockBuilder**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** leveldb **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {FC5E0EA8-920A-44b6-AECE-9EC02DC4F72E}**Custom Properties**

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public BlockBuilder	Private options_ Options	

Attributes

Attribute	Notes	Constraints and tags
buffer_ std::string Private	Destination buffer	<i>Default:</i>
counter_ int Private	Number of entries emitted since restart	<i>Default:</i>
finished_ bool Private	Has Finish() been called?	<i>Default:</i>
last_key_ std::string Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
options_ Options*		<i>Default:</i>
Private Const		

Attribute	Notes	Constraints and tags
restarts_ std::vector<uint32_t>	Restart points	<i>Default:</i>
Private		

Operations

Method	Notes	Parameters
Add() void Public		Slice& [in] key Slice& [in] value
BlockBuilder() Public		Options* [in] options
BlockBuilder() Private		BlockBuilder& [in] prm1
CurrentSizeEstimate() size_t Public		
empty() bool Public		
Finish() Slice Public		
operator=() void Private		BlockBuilder& [in] prm1
Reset() void Public		

BlockContents**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** leveldb **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {CAE7F758-4A88-46d9-B069-19BB4C65E480}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public BlockContents	Public data Slice	

Attributes

Attribute	Notes	Constraints and tags
cachable bool Public	True iff data can be cached	<i>Default:</i>
data Slice Public	Actual contents of data	<i>Default:</i>
heap_allocated bool Public	True iff caller should delete[] data.data()	<i>Default:</i>

BlockHandle

Type: Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** leveldb **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {A36B83AB-0CD5-4845-9701-D40F8CE2A97F}

BlockHandle is a pointer to the extent of a file that stores a data block or a meta block.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public BlockHandle	Private offset_ uint64_t	
Association Source -> Destination	Public BlockHandle	Private size_ uint64_t	
Association Source -> Destination	Public Footer	Private index_handle_ BlockHandle	
Association Source -> Destination	Public Footer	Private metaindex_handle_ BlockHandle	

Attributes

Attribute	Notes	Constraints and tags
offset_ uint64_t Private		<i>Default:</i>
size_ uint64_t Private		<i>Default:</i>

Operations

Method	Notes	Parameters
BlockHandle() Public		
DecodeFrom() Status Public		Slice* [in] input
EncodeTo() void Public		std::string* [in] dst
offset() uint64_t		

Method	Notes	Parameters
Public		
set_offset() void		uint64_t [in] offset
Public		
set_size() void		uint64_t [in] size
Public		
size() uint64_t		
Public		

<anonymous>

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {F79A8270-0875-4b34-841B-842DB95B2EB2}

Maximum encoding length of a BlockHandle

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
kMaxEncodedLength		<i>Default:</i> 10 + 10
Public		

Cache

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {6BA88C89-581C-4908-ADE2-47E08C8BE590}

Custom Properties

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Options	Public block_cache Cache	
Association Source -> Destination	Public TableCache	Private cache_ Cache	

Attributes

Attribute	Notes	Constraints and tags
rep_ Rep* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Cache() Public		
~Cache() Public		
Cache() Private		Cache& [in] prm1
Pure Erase() void Public		Slice& [in] key
Pure Insert() Handle* Public		Slice& [in] key void* [in] value size_t [in] charge void [in] prm1
Pure Lookup() Handle* Public		Slice& [in] key
LRU_Append() void Private		Handle* [in] e

Method	Notes	Parameters
LRU_Remove() void Private		Handle* [in] e
Pure NewId() uint64_t Public		
operator=() void Private		Cache& [in] prm1
Pure Release() void Public		Handle* [in] handle
Unref() void Private		Handle* [in] e
Pure Value() void* Public		Handle* [in] handle

Handle

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {B1778B6E-445D-4852-B8B2-1D9019454222}

Opaque handle to an entry stored in the cache.

Custom Properties

- isActive = False

Compaction

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {C8D24599-36D6-4353-B8FE-479A9B330AAF}

A Compaction encapsulates information about a compaction.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Compaction	Private edit_ VersionEdit	
Association Source -> Destination	Public Compaction	Private input_version_ Version	
Association Source -> Destination	Public Compaction	Private overlapped_bytes_ int64_t	
Association Source -> Destination	Public Compaction	Private max_output_file_size_ uint64_t	

Attributes

Attribute	Notes	Constraints and tags
edit_VersionEdit Private		<i>Default:</i>
grandparent_index_size_t Private	Index in grandparent_starts_	<i>Default:</i>
grandparents_std::vector<FileMetaData*> Private	State used to check for number of overlapping grandparent files (parent == level_ + 1, grandparent == level_ + 2)	<i>Default:</i>
input_version_Version* Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
inputs_ std::vector<FileMetaData*> Private Collection	The two sets of inputs	<i>Default:</i>
level_ int Private		<i>Default:</i>
level_ptrs_ size_t Private Collection	level_ptrs_ holds indices into input_version_->levels_: our state is that we are positioned at one of the file ranges for each higher level than the ones involved in this compaction (i.e. for all L >= level_ + 2).	<i>Default:</i>
max_output_file_size_ uint64_t Private		<i>Default:</i>
overlapped_bytes_ int64_t Private		<i>Default:</i>
seen_key_ bool Private	Some output key has been seen	<i>Default:</i>

Operations

Method	Notes	Parameters
--------	-------	------------

Method	Notes	Parameters
AddInputDeletions() void Public		VersionEdit* [in] edit
~Compaction() Public		
Compaction() Private		int [in] level
edit() VersionEdit* Public		
input() FileMetaData* Public		int [in] which int [in] i
IsBaseLevelForKey() bool Public		Slice& [in] user_key
IsTrivialMove() bool Public		
level() int Public		
MaxOutputFileSize() uint64_t Public		
num_input_files() int Public		int [in] which
ReleaseInputs() void Public		
ShouldStopBefore() bool Public		Slice& [in] internal_key

Comparator

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {ABA10905-4A80-4833-9C35-8ABA4A9399A6}

A Comparator object provides a total order across slices that are used as keys in an sstable or a database. A Comparator implementation must be thread-safe since leveldb may invoke its methods concurrently from multiple threads.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public SkipList	Private compare_ Comparator	
Association Source -> Destination	Public Options	Public comparator Comparator	

Operations

Method	Notes	Parameters
~Comparator() Public		
Pure Compare() int Public		Slice& [in] a Slice& [in] b
Pure FindShortestSeparator() void Public		std::string* [in] start Slice& [in] limit
Pure FindShortSuccessor() void Public		std::string* [in] key
Const Pure Name() char* Public		

CompressionType

Type:**Enumeration****Status:**

Proposed. Version 1.0. Phase 1.0.

Package:leveldb **Keywords:****Detail:**

Created on 8/30/2015. Last modified on 8/30/2015.

GUID:

{05A1A988-1791-4399-B60B-74BF8B724441}

DB contents are stored in a set of blocks, each of which holds a sequence of key,value pairs. Each block may be compressed before being stored in a file. The following enum describes which compression method (if any) is used to compress a block.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Options	Public compression CompressionType	

Connector	Source	Target	Notes

Attributes

Attribute	Notes	Constraints and tags
kNoCompression Public	NOTE: do not change the values of existing entries, as these are part of the persistent format on disk.	<i>Default:</i> 0x0
kSnappyCompression Public		<i>Default:</i> 0x1

DB

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A74C66E6-7ED8-4ff8-A6DD-640EE0A422A1}

A DB is a persistent ordered map from keys to values. A DB is safe for concurrent access from multiple threads without any external synchronization.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public DBImpl	Public DB	
Association Source -> Destination	Public CLevelDBWrapper	Private pdb DB	

Operations

Method	Notes	Parameters

Method	Notes	Parameters
Pure CompactRange() void Public		Slice* [in] begin Slice* [in] end
DB() Public		
~DB() Public		
DB() Private		DB& [in] prm1
Pure Delete() Status Public		WriteOptions& [in] options Slice& [in] key
Pure Get() Status Public		ReadOptions& [in] options Slice& [in] key std::string* [in] value
Pure GetApproximateSizes() void Public		Range* [in] range int [in] n uint64_t* [in] sizes
Pure GetProperty() bool Public		Slice& [in] property std::string* [in] value
Const Pure GetSnapshot() Snapshot* Public		
Pure NewIterator() Iterator* Public		ReadOptions& [in] options
Static Open() Status Public		Options& [in] options std::string& [in] name DB** [in] dbptr
operator=() void Private		DB& [in] prm1
Pure Put() Status Public		WriteOptions& [in] options Slice& [in] key

Method	Notes	Parameters
		Slice& [in] value
Pure ReleaseSnapshot() void Public		Snapshot* [in] snapshot
Pure Write() Status Public		WriteOptions& [in] options WriteBatch* [in] updates

DBImpl

Type: Class **DB**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {95B949E6-652C-4551-A88D-768A680212FD}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private table_cache_ TableCache	
Association Source -> Destination	Public DBImpl	Private tmp_batch_ WriteBatch	
Association Source -> Destination	Public DBImpl	Private versions_ VersionSet	
Generalization Source -> Destination	Public DBImpl	Public DB	
Association Source -> Destination	Public DBImpl	Private snapshots_ SnapshotList	
Association Source -> Destination	Public DBImpl	Private shutting_down_ AtomicPointer	
Association Source -> Destination	Public DBImpl	Private db_lock_ FileLock	

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private stats_ CompactionStats	
Association Source -> Destination	Public DBImpl	Private bg_error_ Status	
Association Source -> Destination	Public DBImpl	Private env_ Env	
Association Source -> Destination	Public DBImpl	Private has_imm_ AtomicPointer	
Association Source -> Destination	Public DBImpl	Private imm_ MemTable	
Association Source -> Destination	Public DBImpl	Private log_ Writer	
Association Source -> Destination	Public DBImpl	Private logfile_ WritableFile	
Association Source -> Destination	Public DBImpl	Private logfile_number_ uint64_t	
Association Source -> Destination	Public DBImpl	Private manual_compaction_ ManualCompaction	
Association Source -> Destination	Public DBImpl	Private mem_ MemTable	
Association Source -> Destination	Public DBImpl	Private mutex_ Mutex	
Association Source -> Destination	Public DBImpl	Private options_ Options	
Association Source -> Destination	Public DBImpl	Private seed_ uint32_t	
Association Source -> Destination	Public DBImpl	Private bg_cv_ CondVar	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
bg_compaction_schedule d_bool Private	Has a background compaction been scheduled or is running?	<i>Default:</i>
bg_cv_ port::CondVar Private	Signalled when background work finishes	<i>Default:</i>
bg_error_ Status Private	Have we encountered a background error in paranoid mode?	<i>Default:</i>
db_lock_ FileLock* Private	Lock over the persistent DB state. Non-NULL iff successfully acquired.	<i>Default:</i>
dbname_ std::string Private Const		<i>Default:</i>
env_ Env* Private	Constant after construction	<i>Default:</i>

Attribute	Notes	Constraints and tags
has_imm_ port::AtomicPointer Private	So bg thread can detect non-NULL imm_ 	<i>Default:</i>
imm_ MemTable* Private	Memtable being compacted 	<i>Default:</i>
internal_comparator_ InternalKeyComparator Private Const	 	<i>Default:</i>
internal_filter_policy_ InternalFilterPolicy Private Const	 	<i>Default:</i>
log_ log::Writer* Private	 	<i>Default:</i>
logfile_ WritableFile* Private	 	<i>Default:</i>

Attribute	Notes	Constraints and tags
logfile_number_ uint64_t Private		<i>Default:</i>
manual_compaction_ ManualCompaction* Private		<i>Default:</i>
mem_ MemTable* Private		<i>Default:</i>
mutex_ port::Mutex Private	State below is protected by mutex_	<i>Default:</i>
options_ Options Private Const	options_.comparator == &internal_comparator_	<i>Default:</i>
owns_cache_ bool Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
owns_info_log_ bool Private		<i>Default:</i>
pending_outputs_ std::set<uint64_t> Private	Set of table files to protect from deletion because they are part of ongoing compactions.	<i>Default:</i>
seed_ uint32_t Private	For sampling.	<i>Default:</i>
shutting_down_ port::AtomicPointer Private		<i>Default:</i>
snapshots_ SnapshotList Private		<i>Default:</i>
stats_ CompactionStats Private Collection		<i>Default:</i>

Attribute	Notes	Constraints and tags
table_cache_ TableCache*	table_cache_ provides its own synchronization	<i>Default:</i>
Private		
tmp_batch_ WriteBatch*		<i>Default:</i>
Private		
versions_ VersionSet*		<i>Default:</i>
Private		
writers_ std::deque<Writer*>	Queue of writers.	<i>Default:</i>
Private		

Operations

Method	Notes	Parameters
BackgroundCall() void		
Private		
BackgroundCompaction() void		
Private		
Static BGWork() void		void* [in] db
Private		
BuildBatchGroup() WriteBatch*		Writer** [in] last_writer
Private		
CleanupCompaction() void		CompactionState* [in] compact
Private		
CompactMemTable() void		
Private		

Method	Notes	Parameters
Private		
FinishCompactionOutputFile() Status		CompactionState* [in] compact Iterator* [in] input
Private		
Get() Status		ReadOptions& [in] options
Public		Slice& [in] key std::string* [in] value
GetApproximateSizes()		Range* [in] range
void		int [in] n
Public		uint64_t* [in] sizes
GetProperty() bool		Slice& [in] property
Public		std::string* [in] value
Const GetSnapshot()		
Snapshot*		
Public		
InstallCompactionResults()		CompactionState* [in] compact
Status		
Private		
MakeRoomForWrite()		bool [in] force
Status		
Private		
MaybeIgnoreError() void		Status* [in] s
Private		
MaybeScheduleCompaction() void		
Private		
NewDB() Status		
Private		
NewInternalIterator()		ReadOptions& [in] prm1
Iterator*		SequenceNumber* [in] latest_snapshot
Private		uint32_t* [in] seed
NewIterator() Iterator*		ReadOptions& [in] prm1
Public		
OpenCompactionOutputFile() Status		CompactionState* [in] compact

Method	Notes	Parameters
Private		
operator=() void		DBImpl& [in] prm1
Private		
Put() Status		WriteOptions& [in] prm1
Public		Slice& [in] key
		Slice& [in] value
RecordBackgroundError () void		Status& [in] s
Private		
RecordReadSample()		Slice [in] key
void		
Public		
Recover() Status		VersionEdit* [in] edit
Private		
RecoverLogFile() Status		uint64_t [in] log_number
Private		VersionEdit* [in] edit
		SequenceNumber* [in] max_sequence
ReleaseSnapshot() void		Snapshot* [in] snapshot
Public		
TEST_CompactMemTab le() Status		
Public		
TEST_CompactRange()		int [in] level
void		Slice* [in] begin
Public		Slice* [in] end
TEST_MaxNextLevelOve rlappingBytes() int64_t		
Public		
TEST_NewInternalIterat or() Iterator*		
Public		
Const user_comparator()		
Comparator*		
Private		
Write() Status		WriteOptions& [in] options
Public		WriteBatch* [in] updates

Method	Notes	Parameters
WriteLevel0Table() Status Private		MemTable* [in] mem VersionEdit* [in] edit Version* [in] base

CompactionStats

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {15CA27BF-EB41-429b-999B-E4108EDF88EC}

Per level compaction stats. stats_[level] stores the stats for compactions that produced data for the specified "level".

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CompactionStats	Public bytes_read int64_t	
Association Source -> Destination	Public CompactionStats	Public bytes_written int64_t	
Association Source -> Destination	Public CompactionStats	Public micros int64_t	
Association Source -> Destination	Public DBImpl	Private stats_ CompactionStats	

Attributes

Attribute	Notes	Constraints and tags
bytes_read int64_t Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
bytes_written int64_t Public		<i>Default:</i>
micros int64_t Public		<i>Default:</i>

Operations

Method	Notes	Parameters
Add() void Public		CompactionStats& [in] c
CompactionStats() Public		

ManualCompaction**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** leveldb **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {F0100D81-1714-4c81-BFD1-DF3CB6366D5A}

Information for a manual compaction

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private manual_compaction_ ManualCompaction	

Attributes

Attribute	Notes	Constraints and tags
begin InternalKey* Public Const	NULL means beginning of key range	<i>Default:</i>
done bool Public		<i>Default:</i>
end InternalKey* Public Const	NULL means end of key range	<i>Default:</i>
level int Public		<i>Default:</i>
tmp_storage InternalKey Public	Used to keep track of compaction progress	<i>Default:</i>

Env

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {AF06C493-FA5E-4fcc-92CE-BFD6E564DCE8}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private env_ Env	
Association Source -> Destination	Public EnvWrapper	Private target_ Env	
Generalization Source -> Destination	Public EnvWrapper	Public Env	
Association Source -> Destination	Public Options	Public env Env	
Association Source -> Destination	Public TableCache	Private env_ Env	
Association Source -> Destination	Public VersionSet	Private env_ Env	
Association Source -> Destination	Public CLevelDBWrapper	Private penv Env	

Operations

Method	Notes	Parameters
Pure CreateDir() Status Public		std::string& [in] dirname
Static Default() Env* Public		
Pure DeleteDir() Status Public		std::string& [in] dirname
Pure DeleteFile() Status Public		std::string& [in] fname
Env() Public		
~Env() Public		
Env() Private		Env& [in] prm1
Pure FileExists() bool Public		std::string& [in] fname
Pure GetChildren() Status		std::string& [in] dir

Method	Notes	Parameters
Public		std::vector<std::string>* [in] result
Pure GetFileSize() Status Public		std::string& [in] fname uint64_t* [in] file_size
Pure GetTestDirectory() Status Public		std::string* [in] path
Pure LockFile() Status Public		std::string& [in] fname FileLock** [in] lock
Pure NewLogger() Status Public		std::string& [in] fname Logger** [in] result
Pure NewRandomAccessFile() Status Public		std::string& [in] fname RandomAccessFile** [in] result
Pure NewSequentialFile() Status Public		std::string& [in] fname SequentialFile** [in] result
Pure NewWritableFile() Status Public		std::string& [in] fname WritableFile** [in] result
Pure NowMicros() uint64_t Public		
operator=() void Private		Env& [in] prm1
Pure RenameFile() Status Public		std::string& [in] src std::string& [in] target
Pure Schedule() void Public		void [in] prm1 void* [in] arg
Pure		int [in] micros

Method	Notes	Parameters
SleepForMicroseconds() void Public		
Pure StartThread() void Public		void [in] prm1 void* [in] arg
Pure UnlockFile() Status Public		FileLock* [in] lock

EnvWrapper

Type:

Class

Env
Proposed. Version 1.0. Phase 1.0.

Status:

Package: leveldb

Keywords:

Detail:

Created on 8/30/2015. Last modified on 8/30/2015.

GUID:

{41237C9D-51C2-4ea0-AC52-DDE90D86B82C}

An implementation of Env that forwards all calls to another Env. May be useful to clients who wish to override just part of the functionality of another Env.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public ErrorEnv	Public EnvWrapper	
Association Source -> Destination	Public EnvWrapper	Private target_ Env	
Generalization Source -> Destination	Public EnvWrapper	Public Env	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
target_ Env* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CreateDir() Status Public		std::string& [in] d
DeleteDir() Status Public		std::string& [in] d
DeleteFile() Status Public		std::string& [in] f
EnvWrapper() Public		Env* [in] t
~EnvWrapper() Public		
FileExists() bool Public		std::string& [in] f
GetChildren() Status Public		std::string& [in] dir std::vector<std::string>* [in] r
GetFileSize() Status Public		std::string& [in] f uint64_t* [in] s
GetTestDirectory() Status Public		std::string* [in] path
LockFile() Status Public		std::string& [in] f FileLock** [in] l
NewLogger() Status Public		std::string& [in] fname Logger** [in] result
NewRandomAccessFile() Status		std::string& [in] f

Method	Notes	Parameters
Public		RandomAccessFile** [in] r
NewSequentialFile() Status Public		std::string& [in] f SequentialFile** [in] r
NewWritableFile() Status Public		std::string& [in] f WritableFile** [in] r
NowMicros() uint64_t Public		
RenameFile() Status Public		std::string& [in] s std::string& [in] t
Schedule() void Public		void [in] prm1 void* [in] a
SleepForMicroseconds() void Public		int [in] micros
StartThread() void Public		void [in] prm1 void* [in] a
target() Env* Public		
UnlockFile() Status Public		FileLock* [in] l

FileLock

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {4C1DFCFA-CC1D-4968-8B8F-7F9BD50A3A77}

Identifies a locked file.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private db_lock_ FileLock	

Operations

Method	Notes	Parameters
FileLock() Public		
~FileLock() Public		
FileLock() Private		FileLock& [in] prm1
operator=() void Private		FileLock& [in] prm1

FileMetaData

Type: Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** leveldb **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {C2804855-4AD0-43e7-97FE-5B0AE44B31F2}**Custom Properties**

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public FileMetaData	Public file_size uint64_t	
Association Source -> Destination	Public FileMetaData	Public number uint64_t	
Association Source -> Destination	Public Version	Private file_to_compact_ FileMetaData	
Association Source -> Destination	Public GetStats	Public seek_file FileMetaData	

Connector	Source	Target	Notes

Attributes

Attribute	Notes	Constraints and tags
allowed_seeks int Public	Seeks allowed until compaction	<i>Default:</i>
file_size uint64_t Public	File size in bytes	<i>Default:</i>
largest InternalKey Public	Largest internal key served by table	<i>Default:</i>
number uint64_t Public		<i>Default:</i>
refs int Public		<i>Default:</i>
smallest InternalKey Public	Smallest internal key served by table	<i>Default:</i>

Operations

Method	Notes	Parameters
FileMetaData() Public		

FileType*Type:***Enumeration***Status:*

Proposed. Version 1.0. Phase 1.0.

Package:

leveldb

Keywords:*Detail:*

Created on 8/30/2015. Last modified on 8/30/2015.

GUID:

{BED3ABB0-4208-46f8-A180-970A7F5729C9}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
kLogFile Public		<i>Default:</i>
kDBLockFile Public		<i>Default:</i>
kTableFile Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
kDescriptorFile Public		<i>Default:</i>
kCurrentFile Public		<i>Default:</i>
kTempFile Public		<i>Default:</i>
kInfoLogFile Public	Either the current one, or an old one	<i>Default:</i>

FilterBlockBuilder

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {8B37EA42-0CE4-4467-A37E-16B8EE05BF45}

A FilterBlockBuilder is used to construct all of the filters for a particular Table. It generates a single string which is stored as a special block in the Table. The sequence of calls to FilterBlockBuilder must match the regexp:
 (StartBlock AddKey*)* Finish

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public FilterBlockBuilder	Private policy_ FilterPolicy	

Attributes

Attribute	Notes	Constraints and tags
filter_offsets_ std::vector<uint32_t> Private		<i>Default:</i>
keys_ std::string Private	Flattened key contents	<i>Default:</i>
policy_ FilterPolicy*		<i>Default:</i>
result_ std::string Private	Filter data computed so far	<i>Default:</i>
start_ std::vector<size_t> Private	Starting index in keys_ of each key	<i>Default:</i>

Attribute	Notes	Constraints and tags
tmp_keys_ std::vector<Slice> Private	policy_->CreateFilter() argument	<i>Default:</i>

Operations

Method	Notes	Parameters
AddKey() void Public		Slice& [in] key
FilterBlockBuilder() Public		FilterPolicy* [in] prm1
FilterBlockBuilder() Private		FilterBlockBuilder& [in] prm1
Finish() Slice Public		
GenerateFilter() void Private		
operator=() void Private		FilterBlockBuilder& [in] prm1
StartBlock() void Public		uint64_t [in] block_offset

FilterBlockReader

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {00658227-114C-4135-9BC2-DDCAB1F0B1C9}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association	Public	Private policy_-	

Connector	Source	Target	Notes
Source -> Destination	FilterBlockReader	FilterPolicy	

Attributes

Attribute	Notes	Constraints and tags
base_lg_size_t Private	Encoding parameter (see kFilterBaseLg in .cc file)	<i>Default:</i>
data_char* Private Const	Pointer to filter data (at block-start)	<i>Default:</i>
num_size_t Private	Number of entries in offset array	<i>Default:</i>
offset_char* Private Const	Pointer to beginning of offset array (at block-end)	<i>Default:</i>
policy_FilterPolicy* Private Const		<i>Default:</i>

Operations

Method	Notes	Parameters
FilterBlockReader() Public		FilterPolicy* [in] policy Slice& [in] contents

Method	Notes	Parameters
KeyMayMatch() bool Public		uint64_t [in] block_offset Slice& [in] key

FilterPolicy

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {D1D72786-4F01-44dd-BA5C-EC1CE47BC6AB}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public FilterBlockReader	Private policy_FilterPolicy	
Association Source -> Destination	Public FilterBlockBuilder	Private policy_FilterPolicy	
Association Source -> Destination	Public Options	Public filter_policy FilterPolicy	

Operations

Method	Notes	Parameters
Pure CreateFilter() void Public		Slice* [in] keys int [in] n std::string* [in] dst
~FilterPolicy() Public		
Pure KeyMayMatch() bool Public		Slice& [in] key Slice& [in] filter

Method	Notes	Parameters
Const Pure Name() char* Public		

Footer

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb

Keywords:

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {8394ECF7-1C10-4b20-9785-C9230B41A4E2}

Footer encapsulates the fixed information stored at the tail end of every table file.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Footer	Private index_handle_ BlockHandle	
Association Source -> Destination	Public Footer	Private metaindex_handle_ BlockHandle	

Attributes

Attribute	Notes	Constraints and tags
index_handle_ BlockHandle Private		<i>Default:</i>
metaindex_handle_ BlockHandle Private		<i>Default:</i>

Operations

Method	Notes	Parameters
DecodeFrom() Status Public		Slice* [in] input
EncodeTo() void Public		std::string* [in] dst
Footer() Public		
Const index_handle() BlockHandle& Public		
Const metaindex_handle() BlockHandle& Public		
set_index_handle() void Public		BlockHandle& [in] h
set_metaindex_handle() void Public		BlockHandle& [in] h

<anonymous>

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {9CA38FC7-2145-460b-80AD-1C3457076E27}

Encoded length of a Footer. Note that the serialization of a Footer will always occupy exactly this many bytes. It consists of two block handles and a magic number.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
kEncodedLength Public		<i>Default:</i> $2 * \text{BlockHandle::kMaxEncodedLength} + 8$

Histogram

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {90856684-FAA7-467b-A1A6-04040CAA6B20}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
buckets_ double Private Collection		<i>Default:</i>
kBucketLimit double Private Static Const Collection		<i>Default:</i>
max_ double Private		<i>Default:</i>
min_ double Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
num_ double Private		<i>Default:</i>
sum_ double Private		<i>Default:</i>
sum_squares_ double Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Add() void Public		double [in] value
Average() double Private		
Clear() void Public		
Histogram() Public		
~Histogram() Public		
Median() double Private		
Merge() void Public		Histogram& [in] other
Percentile() double Private		double [in] p
StandardDeviation() double Private		
ToString() std::string Public		

<anonymous>

Type: **Enumeration**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {0B8D1958-45FC-4572-8895-BC761A90B20C}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
kNumBuckets Private		<i>Default: 154</i>

Iterator

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {052E0EA8-FA48-4d98-99CF-1068CF13876E}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public IteratorWrapper	Private iter_ Iterator	
Association Source -> Destination	Public Iterator	Private cleanup_ Cleanup	

Attributes

Attribute	Notes	Constraints and tags
cleanup_ Cleanup Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Iterator() Public		
~Iterator() Public		
Iterator() Private		Iterator& [in] prm1
Pure key() Slice Public		
Pure Next() void Public		
operator=() void Private		Iterator& [in] prm1
Pure Prev() void Public		
RegisterCleanup() void Public		CleanupFunction [in] function void* [in] arg1 void* [in] arg2
Pure Seek() void Public		Slice& [in] target
Pure SeekToFirst() void Public		
Pure SeekToLast() void Public		
Pure status() Status Public		
Pure Valid() bool Public		
Pure value() Slice Public		

Cleanup

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {17C02D3E-56DD-4a3a-B9E6-001A129AB913}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Iterator	Private cleanup_ Cleanup	
Association Source -> Destination	Public Cleanup	Public next Cleanup	

Attributes

Attribute	Notes	Constraints and tags
arg1 void* Public		<i>Default:</i>
arg2 void* Public		<i>Default:</i>
function CleanupFunction Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
next Cleanup*		<i>Default:</i>

IteratorWrapper

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {E7D4922F-71FB-4d69-86B9-370820A4BA71}

A internal wrapper class with an interface similar to Iterator that caches the valid() and key() results for an underlying iterator. This can help avoid virtual function calls and also gives better cache locality.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public IteratorWrapper	Private iter_ Iterator	
Association Source -> Destination	Public IteratorWrapper	Private key_ Slice	

Attributes

Attribute	Notes	Constraints and tags
iter_ Iterator*		<i>Default:</i>

Attribute	Notes	Constraints and tags
key_ Slice Private		<i>Default:</i>
valid_ bool Private		<i>Default:</i>

Operations

Method	Notes	Parameters
iter() Iterator*		
Public		
IteratorWrapper()		
Public		
IteratorWrapper()		Iterator* [in] iter
Public		
~IteratorWrapper()		
Public		
key() Slice		
Public		
Next() void		
Public		
Prev() void		
Public		
Seek() void		Slice& [in] k
Public		
SeekToFirst() void		
Public		
SeekToLast() void		
Public		
Set() void		Iterator* [in] iter
Public		
status() Status		
Public		
Update() void		
Private		
Valid() bool		
Public		
value() Slice		
Public		

Logger

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {0292D029-F2CA-43f0-9DAE-61FBB290815C}

An interface for writing log messages.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public PosixLogger	Public Logger	
Association Source -> Destination	Public Options	Public info_log Logger	

Operations

Method	Notes	Parameters
Logger() Public		
~Logger() Public		
Logger() Private		Logger& [in] prm1
Pure Logv() void Public		char* [in] format va_list [in] ap
operator=() void Private		Logger& [in] prm1

MemTable

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {58BBBCE9-B971-496a-B7E9-6ED75E127361}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public MemTable	Private arena_ Arena	
Association Source -> Destination	Public MemTable	Private comparator_ KeyComparator	
Association Source -> Destination	Public MemTable	Private table_ Table	
Association Source -> Destination	Public DBImpl	Private imm_ MemTable	
Association Source -> Destination	Public DBImpl	Private mem_ MemTable	

Attributes

Attribute	Notes	Constraints and tags
arena_ Arena Private		<i>Default:</i>
comparator_ KeyComparator Private		<i>Default:</i>
refs_ int Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
table_ Table Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Add() void Public		SequenceNumber [in] seq ValueType [in] type Slice& [in] key Slice& [in] value
ApproximateMemoryUsa ge() size_t Public		
Get() bool Public		LookupKey& [in] key std::string* [in] value Status* [in] s
MemTable() Public		InternalKeyComparator& [in] comparator
~MemTable() Private		
MemTable() Private		MemTable& [in] prm1
NewIterator() Iterator* Public		
operator=() void Private		MemTable& [in] prm1
Ref() void Public		
Unref() void Public		

KeyComparatorType: Class

Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {19C6A663-660F-4918-88EB-BD9741226361}

Private since only Unref() should be used to delete it

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public MemTable	Private comparator_ KeyComparator	

Attributes

Attribute	Notes	Constraints and tags
comparator InternalKeyComparator Public Const		<i>Default:</i>

Operations

Method	Notes	Parameters
KeyComparator() Public		InternalKeyComparator& [in] c
operator() int Public		char* [in] a char* [in] b

Table

Type: **Class** SkipList<const char*, KeyComparator>
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {03E1BB07-E694-4c34-85A1-B94C44293C6E}

Custom Properties

- isActive = False

Node**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** leveldb **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {1EC84743-B026-4764-B07C-762F664A6D00}

Implementation details follow

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public SkipList	Private head_ Node	
Association Source -> Destination	Public Iterator	Private node_ Node	
Association Source -> Destination	Public Node	Private next_ AtomicPointer	

Attributes

Attribute	Notes	Constraints and tags
key Key Public		<i>Default:</i>

Attribute	Notes	Constraints and tags
next_ port::AtomicPointer Private Collection	Array of length equal to the node height. next_[0] is lowest level link.	<i>Default:</i>

Operations

Method	Notes	Parameters
Next() Node*		int [in] n
Public		
NoBarrier_Next() Node*		int [in] n
Public		
NoBarrier_SetNext() void		int [in] n
Public		Node* [in] x
Node()		Key& [in] k
Public		
SetNext() void		int [in] n
Public		Node* [in] x

Options

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D78451D4-6D72-4007-8A46-53632472400A}

Options to control the behavior of a database (passed to DB::Open)

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association	Public	Private options_	

Connector	Source	Target	Notes
Source -> Destination	DBImpl	Options	
Association Source -> Destination	Public VersionSet	Private options_ Options	
Association Source -> Destination	Public Options	Public block_cache Cache	
Association Source -> Destination	Public Options	Public comparator Comparator	
Association Source -> Destination	Public Options	Public compression CompressionType	
Association Source -> Destination	Public Options	Public env Env	
Association Source -> Destination	Public Options	Public filter_policy FilterPolicy	
Association Source -> Destination	Public Options	Public info_log Logger	
Association Source -> Destination	Public BlockBuilder	Private options_ Options	
Association Source -> Destination	Public TableCache	Private options_ Options	
Association Source -> Destination	Public CLevelDBWrapper	Private options Options	

Attributes

Attribute	Notes	Constraints and tags
block_cache Cache* Public	If non-NULL, use the specified cache for blocks. If NULL, leveldb will automatically create and use an 8MB internal cache. Default: NULL	<i>Default:</i>
block_restart_interval int Public	Number of keys between restart points for delta encoding of keys. This parameter can be changed dynamically. Most clients should leave this parameter alone. Default: 16	<i>Default:</i>

Attribute	Notes	Constraints and tags
block_size size_t Public	Approximate size of user data packed per block. Note that the block size specified here corresponds to uncompressed data. The actual size of the unit read from disk may be smaller if compression is enabled. This parameter can be changed dynamically. Default: 4K	<i>Default:</i>
comparator Comparator* Public Const	Comparator used to define the order of keys in the table. Default: a comparator that uses lexicographic byte-wise ordering REQUIRES: The client must ensure that the comparator supplied here has the same name and orders keys *exactly* the same as the comparator provided to previous open calls on the same DB.	<i>Default:</i>
compression CompressionType Public	Compress blocks using the specified compression algorithm. This parameter can be changed dynamically. Default: kSnappyCompression, which gives lightweight but fast compression. Typical speeds of kSnappyCompression on an Intel(R) Core(TM)2 2.4GHz: ~200-500MB/s compression ~400-800MB/s decompression Note that these speeds are significantly faster than most persistent storage speeds, and therefore it is typically never worth switching to kNoCompression. Even if the input data is incompressible, the kSnappyCompression implementation will efficiently detect that and will switch to uncompressed mode.	<i>Default:</i>
create_if_missing bool Public	If true, the database will be created if it is missing. Default: false	<i>Default:</i>
env Env* Public	Use the specified object to interact with the environment, e.g. to read/write files, schedule background work, etc. Default: Env::Default()	<i>Default:</i>

Attribute	Notes	Constraints and tags
error_if_exists bool Public	If true, an error is raised if the database already exists. Default: false	<i>Default:</i>
filter_policy FilterPolicy* Public Const	If non-NULL, use the specified filter policy to reduce disk reads. Many applications will benefit from passing the result of NewBloomFilterPolicy() here. Default: NULL	<i>Default:</i>
info_log Logger* Public	Any internal progress/error information generated by the db will be written to info_log if it is non-NULL, or to a file stored in the same directory as the DB contents if info_log is NULL. Default: NULL	<i>Default:</i>
max_open_files int Public	Number of open files that can be used by the DB. You may need to increase this if your database has a large working set (budget one open file per 2MB of working set). Default: 1000	<i>Default:</i>
paranoid_checks bool Public	If true, the implementation will do aggressive checking of the data it is processing and will stop early if it detects any errors. This may have unforeseen ramifications: for example, a corruption of one DB entry may cause a large number of entries to become unreadable or for the entire DB to become unopenable. Default: false	<i>Default:</i>
write_buffer_size size_t Public	Amount of data to build up in memory (backed by an unsorted log on disk) before converting to a sorted on-disk file. Larger values increase performance, especially during bulk loads. Up to two write buffers may be held in memory at the same time, so you may wish to adjust this parameter to control memory usage. Also, a larger write buffer will result in a longer recovery time the next time the database is opened. Default: 4MB	<i>Default:</i>

[*Operations*](#)

Method	Notes	Parameters
Options() Public		

PosixLogger

Type: Class **Logger**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {972CFF9A-2CDE-467e-B1D7-7ECAD951AA1B}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public PosixLogger	Public Logger	

Attributes

Attribute	Notes	Constraints and tags
file_ FILE* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Logv() void Public		char* [in] format va_list [in] ap
PosixLogger() Public		FILE* [in] f uint64_t [in] prm1
~PosixLogger()		

Method	Notes	Parameters
Public		

Random

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {4690055D-CFDA-4de4-B95F-65CD22DD75C0}

A very simple random number generator. Not especially good at generating truly random bits, but good enough for our needs in this package.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Random	Private seed_ uint32_t	
Association Source -> Destination	Public SkipList	Private rnd_ Random	

Attributes

Attribute	Notes	Constraints and tags
seed_ uint32_t Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Next() uint32_t Public		
OneIn() bool Public		int [in] n
Random() Public		uint32_t [in] s

Method	Notes	Parameters
Skewed() uint32_t Public		int [in] max_log
Uniform() uint32_t Public		int [in] n

RandomAccessFile

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb Keywords:

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {FE478C3F-C01B-49b2-A977-3378A39DD694}

A file abstraction for randomly reading the contents of a file.

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
operator=() void Private		RandomAccessFile& [in] prm1
RandomAccessFile() Public		
~RandomAccessFile() Public		
RandomAccessFile() Private		RandomAccessFile& [in] prm1
Pure Read() Status Public		uint64_t [in] offset size_t [in] n Slice* [in] result char* [in] scratch

Range

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {2FBF7528-9F4F-4a5f-82C3-01AACFCEF93A}

A range of keys

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Range	Public limit Slice	
Association Source -> Destination	Public Range	Public start Slice	

Attributes

Attribute	Notes	Constraints and tags
limit Slice Public	Not included in the range	<i>Default:</i>
start Slice Public	Included in the range	<i>Default:</i>

Operations

Method	Notes	Parameters
Range() Public		
Range() Public		Slice& [in] s Slice& [in] l

ReadOptions

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {897D7509-F5D7-479f-9505-8E4DC4FFB11C}

Options that control read operations

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public ReadOptions	Public snapshot Snapshot	
Association Source -> Destination	Public CLevelDBWrapper	Private iteroptions ReadOptions	
Association Source -> Destination	Public CLevelDBWrapper	Private readoptions ReadOptions	

Attributes

Attribute	Notes	Constraints and tags
fill_cache bool Public	Should the data read for this iteration be cached in memory? Callers may wish to set this field to false for bulk scans. Default: true	<i>Default:</i>
snapshot Snapshot* Public Const	If "snapshot" is non-NULL, read as of the supplied snapshot (which must belong to the DB that is being read and which must not have been released). If "snapshot" is NULL, use an implicit snapshot of the state at the beginning of this read operation. Default: NULL	<i>Default:</i>

Attribute	Notes	Constraints and tags
verify_checksums bool Public	If true, all data read from underlying storage will be verified against corresponding checksums. Default: false	<i>Default:</i>

Operations

Method	Notes	Parameters
ReadOptions() Public		

SequentialFile

Type: Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** leveldb **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {3F9F1B5F-C59A-4d24-AB33-2C986979E50B}

A file abstraction for reading sequentially through a file

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Reader	Private file_SequentialFile	

Operations

Method	Notes	Parameters
operator=() void Private		SequentialFile& [in] prm1
Pure Read() Status Public		size_t [in] n Slice* [in] result char* [in] scratch
SequentialFile()		

Method	Notes	Parameters
Public		
~SequentialFile() Public		
SequentialFile() Private		SequentialFile& [in] prm1
Pure Skip() Status Public		uint64_t [in] n

SkipList

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {AAC33694-D7EC-42ff-A54E-6141BE621B24}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public SkipList	Private arena_ Arena	
Association Source -> Destination	Public SkipList	Private compare_ Comparator	
Association Source -> Destination	Public SkipList	Private head_ Node	
Association Source -> Destination	Public SkipList	Private max_height_ AtomicPointer	
Association Source -> Destination	Public SkipList	Private rnd_ Random	
Association Source -> Destination	Public Iterator	Private list_ SkipList	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
arena_ Arena* Private	Arena used for allocations of nodes	<i>Default:</i>
compare_ Comparator Private	Immutable after construction	<i>Default:</i>
head_ Node* Private		<i>Default:</i>
max_height_ port::AtomicPointer Private	Height of the entire list	<i>Default:</i>
rnd_ Random Private	Read/written only by Insert().	<i>Default:</i>

Operations

Method	Notes	Parameters
Contains() bool Public		Key& [in] key
Equal() bool Private		Key& [in] a Key& [in] b
FindGreaterOrEqual()		Key& [in] key

Method	Notes	Parameters
Node* Private		Node** [in] prev
FindLast() Node* Private		
FindLessThan() Node* Private		Key& [in] key
GetMaxHeight() int Private		
Insert() void Public		Key& [in] key
KeyIsAfterNode() bool Private		Key& [in] key Node* [in] n
NewNode() Node* Private		Key& [in] key int [in] height
operator=() void Private		SkipList& [in] prm1
RandomHeight() int Private		
SkipList() Public		Comparator [in] cmp Arena* [in] arena
SkipList() Private		SkipList& [in] prm1

<anonymous>

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {2343C260-EB77-483d-B212-AB77563A6454}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
kMaxHeight Private		<i>Default:</i> 12

Iterator

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {810D3A35-913C-4af5-8596-9F67C411ED1D}

Iteration over the contents of a skip list

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Iterator	Private list_ SkipList	
Association Source -> Destination	Public Iterator	Private node_ Node	

Attributes

Attribute	Notes	Constraints and tags
list_ SkipList* Private Const		<i>Default:</i>

Attribute	Notes	Constraints and tags
node_ Node* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Iterator() Public		SkipList* [in] list
Const key() Key& Public		
Next() void Public		
Prev() void Public		
Seek() void Public		Key& [in] target
SeekToFirst() void Public		
SeekToLast() void Public		
Valid() bool Public		

Slice

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {8AC64DFE-5840-4241-AEA7-74DB99E8B6E8}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public BlockContents	Public data Slice	

Connector	Source	Target	Notes
Association Source -> Destination	Public IteratorWrapper	Private key_ Slice	
Association Source -> Destination	Public Reader	Private buffer_ Slice	
Association Source -> Destination	Public Range	Public limit Slice	
Association Source -> Destination	Public Range	Public start Slice	

Attributes

Attribute	Notes	Constraints and tags
data_ char* Private Const		<i>Default:</i>
size_ size_t Private		<i>Default:</i>

Operations

Method	Notes	Parameters
clear() void Public		
compare() int Public		Slice& [in] b
Const data() char* Public		
empty() bool Public		
operator[]() char Public		size_t [in] n
remove_prefix() void Public		size_t [in] n
size() size_t		

Method	Notes	Parameters
Public		
Slice() Public		
Slice() Public		char* [in] d size_t [in] n
Slice() Public		std::string& [in] s
Slice() Public		char* [in] s
starts_with() bool Public		Slice& [in] x
ToString() std::string Public		

Snapshot

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {3877F03D-DD8E-4189-A257-43DE158337DA}

Abstract handle to particular state of a DB. A Snapshot is an immutable object and can therefore be safely accessed from multiple threads without any external synchronization.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public ReadOptions	Public snapshot Snapshot	
Generalization Source -> Destination	Public SnapshotImpl	Public Snapshot	

Operations

Method	Notes	Parameters
~Snapshot() Protected		

SnapshotImpl

Type:	Class	Snapshot
Status:	Proposed.	Version 1.0. Phase 1.0.
Package:	leveldb	Keywords:
Detail:	Created on 8/30/2015. Last modified on 8/30/2015.	
GUID:	{1C453660-E570-4184-B990-06688C420432}	

Snapshots are kept in a doubly-linked list in the DB. Each SnapshotImpl corresponds to a particular sequence number.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public SnapshotImpl	Private list_ SnapshotList	
Association Source -> Destination	Public SnapshotImpl	Private next_ SnapshotImpl	
Association Source -> Destination	Public SnapshotImpl	Private prev_ SnapshotImpl	
Generalization Source -> Destination	Public SnapshotImpl	Public Snapshot	
Association Source -> Destination	Public SnapshotList	Private list_ SnapshotImpl	

Attributes

Attribute	Notes	Constraints and tags
list_ SnapshotList* Private	just for sanity checks	Default:

Attribute	Notes	Constraints and tags
next_ SnapshotImpl* Private		<i>Default:</i>
number_ SequenceNumber Public	const after creation	<i>Default:</i>
prev_ SnapshotImpl* Private	SnapshotImpl is kept in a doubly-linked circular list	<i>Default:</i>

SnapshotList

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {CA3DA173-6C26-4404-B625-1E6CD933B955}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private snapshots_ SnapshotList	
Association Source -> Destination	Public SnapshotImpl	Private list_ SnapshotList	
Association Source -> Destination	Public SnapshotList	Private list_ SnapshotImpl	

Connector	Source	Target	Notes

Attributes

Attribute	Notes	Constraints and tags
list_ SnapshotImpl Private	Dummy head of doubly-linked list of snapshots	<i>Default:</i>

Operations

Method	Notes	Parameters
Delete() void Public		SnapshotImpl* [in] s
empty() bool Public		
Const New() SnapshotImpl* Public		SequenceNumber [in] seq
newest() SnapshotImpl* Public		
oldest() SnapshotImpl* Public		
SnapshotList() Public		

Status

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {7AD0242A-041D-4448-99FA-699CF38EE689}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association	Public	Private bg_error_	

Connector	Source	Target	Notes
Source -> Destination	DBImpl	Status	

Attributes

Attribute	Notes	Constraints and tags
state_ char* Private Const	OK status has a NULL state_. Otherwise, state_ is a new[] array of the following form: state_[0..3] == length of message state_[4] == code state_[5..] == message	<i>Default:</i>

Operations

Method	Notes	Parameters
code() Code Private		
Static Const CopyState() char* Private		char* [in] s
Static Corruption() Status Public		Slice& [in] msg Slice& [in] msg2
Static InvalidArgument() Status Public		Slice& [in] msg Slice& [in] msg2
Static IOError() Status Public		Slice& [in] msg Slice& [in] msg2
IsCorruption() bool Public		
IsIOError() bool Public		
IsNotFound() bool Public		
Static NotFound() Status Public		Slice& [in] msg Slice& [in] msg2
Static NotSupported() Status Public		Slice& [in] msg Slice& [in] msg2

Method	Notes	Parameters
Static OK() Status Public		
ok() bool Public		
operator=() void Public		Status& [in] s
Status() Public		
~Status() Public		
Status() Public		Status& [in] s
Status() Private		Code [in] code Slice& [in] msg Slice& [in] msg2
ToString() std::string Public		

Code

Type:

Enumeration

Status:

Proposed. Version 1.0. Phase 1.0.

Package:

leveldb *Keywords:*

Detail:

Created on 8/30/2015. Last modified on 8/30/2015.

GUID:

{53FF8251-1847-49d0-94B0-82D4E0EA09F5}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
kOk Private		<i>Default:</i> 0

Attribute	Notes	Constraints and tags
kNotFound Private		<i>Default:</i> 1
kCorruption Private		<i>Default:</i> 2
kNotSupported Private		<i>Default:</i> 3
kInvalidArgument Private		<i>Default:</i> 4
kIOError Private		<i>Default:</i> 5

Table

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {76F473F0-862F-430c-8470-3EB1EACEB5B6}

A Table is a sorted map from strings to strings. Tables are immutable and persistent. A Table may be safely

accessed from multiple threads without external synchronization.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
<u>Association</u> Source -> Destination	Public MemTable	Private table_ Table	

Attributes

Attribute	Notes	Constraints and tags
rep_ Rep* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
ApproximateOffsetOf() uint64_t Public		Slice& [in] key
Static BlockReader() Iterator* Private		void* [in] prm1 ReadOptions& [in] prm2 Slice& [in] prm3
InternalGet() Status Private		ReadOptions& [in] prm1 Slice& [in] key void* [in] arg void [in] prm2
NewIterator() Iterator* Public		ReadOptions& [in] prm1
Static Open() Status Public		Options& [in] options RandomAccessFile* [in] file

Method	Notes	Parameters
		uint64_t [in] file_size Table** [in] table
operator=() void Private		Table& [in] prm1
ReadFilter() void Private		Slice& [in] filter_handle_value
ReadMeta() void Private		Footer& [in] footer
~Table() Public		
Table() Private		Rep* [in] rep
Table() Private		Table& [in] prm1

TableBuilder

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D0E61F6E-ADB5-4d69-A5C8-149C1FD38E4D}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
rep_ Rep* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Abandon() void Public		
Add() void Public		Slice& [in] key Slice& [in] value
ChangeOptions() Status Public		Options& [in] options
FileSize() uint64_t Public		
Finish() Status Public		
Flush() void Public		
NumEntries() uint64_t Public		
ok() bool Private		
operator=() void Private		TableBuilder& [in] prm1
status() Status Public		
TableBuilder() Public		Options& [in] options WritableFile* [in] file
~TableBuilder() Public		
TableBuilder() Private		TableBuilder& [in] prm1
WriteBlock() void Private		BlockBuilder* [in] block BlockHandle* [in] handle
WriteRawBlock() void Private		Slice& [in] data CompressionType [in] prm1 BlockHandle* [in] handle

TableCache**Type:****Class****Status:**

Proposed. Version 1.0. Phase 1.0.

Package: leveldb **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {745C3237-A319-4dc1-9EBA-ADABA61B58CF}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private table_cache_ TableCache	
Association Source -> Destination	Public VersionSet	Private table_cache_ TableCache	
Association Source -> Destination	Public TableCache	Private cache_ Cache	
Association Source -> Destination	Public TableCache	Private env_ Env	
Association Source -> Destination	Public TableCache	Private options_ Options	

Attributes

Attribute	Notes	Constraints and tags
cache_ Cache* Private		<i>Default:</i>
dbname_ std::string Private Const		<i>Default:</i>

Attribute	Notes	Constraints and tags
env_ Env* Private		<i>Default:</i>
options_ Options* Private Const		<i>Default:</i>

Operations

Method	Notes	Parameters
Evict() void Public		uint64_t [in] file_number
FindTable() Status Private		uint64_t [in] file_number uint64_t [in] file_size Cache::Handle** [in] prm1
Get() Status Public		ReadOptions& [in] options uint64_t [in] file_number uint64_t [in] file_size Slice& [in] k void* [in] arg void [in] prm1
NewIterator() Iterator* Public		ReadOptions& [in] options uint64_t [in] file_number uint64_t [in] file_size Table** [in] tableptr
TableCache() Public		std::string& [in] dbname

Method	Notes	Parameters
		Options* [in] options int [in] entries
~TableCache() Public		

Version

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {A59798BE-8706-4266-A449-4E857A3DC815}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Compaction	Private input_version_Version	
Association Source -> Destination	Public Version	Private file_to_compact_FileMetaData	
Association Source -> Destination	Public Version	Private next_Version	
Association Source -> Destination	Public Version	Private prev_Version	
Association Source -> Destination	Public Version	Private vset_VersionSet	
Association Source -> Destination	Public VersionSet	Private current_Version	
Association Source -> Destination	Public VersionSet	Private dummy_versions_Version	

Attributes

Attribute	Notes	Constraints and tags
compaction_level_ int Private		<i>Default:</i>
compaction_score_ double Private	Level that should be compacted next and its compaction score. Score < 1 means compaction is not strictly needed. These fields are initialized by Finalize().	<i>Default:</i>
file_to_compact_ FileMetaData* Private	Next file to compact based on seek stats.	<i>Default:</i>
file_to_compact_level_ int Private		<i>Default:</i>
files_ std::vector<FileMetaData*> Private Collection	List of files per level	<i>Default:</i>
next_ Version* Private	Next version in linked list	<i>Default:</i>

Attribute	Notes	Constraints and tags
prev_ Version* Private	Previous version in linked list	<i>Default:</i>
refs_ int Private	Number of live refs to this version	<i>Default:</i>
vset_ VersionSet* Private	VersionSet to which this Version belongs	<i>Default:</i>

Operations

Method	Notes	Parameters
AddIterators() void Public		ReadOptions& [in] prm1 std::vector<Iterator*>* [in] iters
DebugString() std::string Public		
ForEachOverlapping() void Private		Slice [in] user_key Slice [in] internal_key void* [in] arg bool [in] prm1
Get() Status Public		ReadOptions& [in] prm1 LookupKey& [in] key std::string* [in] val GetStats* [in] stats

Method	Notes	Parameters
GetOverlappingInputs() void Public		int [in] level InternalKey* [in] begin InternalKey* [in] end NULL means before all keys std::vector<FileMetaData*>* [in] inputs NULL means after all keys
NewConcatenatingIterator() Iterator* Private		ReadOptions& [in] prm1 int [in] level
NumFiles() int Public		int [in] level
operator=() void Private		Version& [in] prm1
OverlapInLevel() bool Public		int [in] level Slice* [in] smallest_user_key Slice* [in] largest_user_key
PickLevelForMemTableOutput() int Public		Slice& [in] smallest_user_key Slice& [in] largest_user_key
RecordReadSample() bool Public		Slice [in] key
Ref() void Public		
Unref() void Public		
UpdateStats() bool Public		GetStats& [in] stats
Version() Private		VersionSet* [in] vset
~Version() Private		
Version() Private		Version& [in] prm1

GetStats

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {B939A18F-B363-4f69-BA3C-C3F30F6C26AD}

Lookup the value for key. If found, store it in *val and return OK. Else return a non-OK status. Fills *stats.
REQUIRES: lock is not held

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public GetStats	Public seek_file FileMetaData	

Attributes

Attribute	Notes	Constraints and tags
seek_file FileMetaData*		<i>Default:</i>
Public		
seek_file_level int		<i>Default:</i>
Public		

VersionEdit

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {367B9C36-EC29-4f2d-B1D8-86CE436F4950}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Compaction	Private edit_ VersionEdit	
Association Source -> Destination	Public VersionEdit	Private prev_log_number_ uint64_t	
Association Source -> Destination	Public VersionEdit	Private deleted_files_ DeletedFileSet	
Association Source -> Destination	Public VersionEdit	Private next_file_number_ uint64_t	
Association Source -> Destination	Public VersionEdit	Private log_number_ uint64_t	

Attributes

Attribute	Notes	Constraints and tags
compact_pointers_ std::vector< std::pair<int, InternalKey> > Private		<i>Default:</i>
comparator_ std::string Private		<i>Default:</i>
deleted_files_ DeletedFileSet Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
has_comparator_ bool Private		<i>Default:</i>
has_last_sequence_ bool Private		<i>Default:</i>
has_log_number_ bool Private		<i>Default:</i>
has_next_file_number_ bool Private		<i>Default:</i>
has_prev_log_number_ bool Private		<i>Default:</i>
last_sequence_ SequenceNumber Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
log_number_ uint64_t Private		<i>Default:</i>
new_files_ std::vector<std::pair<int, FileMetaData> > Private		<i>Default:</i>
next_file_number_ uint64_t Private		<i>Default:</i>
prev_log_number_ uint64_t Private		<i>Default:</i>

Operations

Method	Notes	Parameters
AddFile() void Public		int [in] level uint64_t [in] file uint64_t [in] file_size InternalKey& [in] smallest InternalKey& [in] largest
Clear() void Public		
DebugString() std::string Public		
DecodeFrom() Status		Slice& [in] src

Method	Notes	Parameters
Public		
DeleteFile() void Public		int [in] level uint64_t [in] file
EncodeTo() void Public		std::string* [in] dst
SetCompactPointer() void Public		int [in] level InternalKey& [in] key
SetComparatorName() void Public		Slice& [in] name
SetLastSequence() void Public		SequenceNumber [in] seq
SetLogNumber() void Public		uint64_t [in] num
SetNextFile() void Public		uint64_t [in] num
SetPrevLogNumber() void Public		uint64_t [in] num
VersionEdit() Public		
~VersionEdit() Public		

DeletedFileSet

Type: Class **std::set< std::pair<int, uint64_t> >**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {AF9D352F-3F1E-4a4f-BFBB-62608CB10B4A}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
-----------	--------	--------	-------

Connector	Source	Target	Notes
Association Source -> Destination	Public VersionEdit	Private deleted_files_ DeletedFileSet	

VersionSet

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {9F5192AF-0B85-4902-9DFF-BF7C0EA60AAC}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private versions_ VersionSet	
Association Source -> Destination	Public VersionSet	Private options_ Options	
Association Source -> Destination	Public VersionSet	Private prev_log_number_ uint64_t	
Association Source -> Destination	Public VersionSet	Private table_cache_ TableCache	
Association Source -> Destination	Public VersionSet	Private log_number_ uint64_t	
Association Source -> Destination	Public VersionSet	Private next_file_number_ uint64_t	
Association Source -> Destination	Public VersionSet	Private manifest_file_number_ uint64_t	
Association Source -> Destination	Public Version	Private vset_ VersionSet	
Association Source -> Destination	Public VersionSet	Private current_ Version	

Connector	Source	Target	Notes
Association Source -> Destination	Public VersionSet	Private descriptor_file_ WritableFile	
Association Source -> Destination	Public VersionSet	Private descriptor_log_ Writer	
Association Source -> Destination	Public VersionSet	Private dummy_versions_ Version	
Association Source -> Destination	Public VersionSet	Private env_ Env	
Association Source -> Destination	Public VersionSet	Private last_sequence_ uint64_t	

Attributes

Attribute	Notes	Constraints and tags
compact_pointer_ std::string Private Collection	Per-level key at which the next compaction at that level should start. Either an empty string, or a valid InternalKey.	<i>Default:</i>
current_ Version* Private	== dummy_versions_.prev_	<i>Default:</i>
dbname_ std::string Private Const		<i>Default:</i>
descriptor_file_ WritableFile* Private	Opened lazily	<i>Default:</i>

Attribute	Notes	Constraints and tags
descriptor_log_ log::Writer* Private		<i>Default:</i>
dummy_versions_ Version Private	Head of circular doubly-linked list of versions.	<i>Default:</i>
env_ Env* Private		<i>Default:</i>
icmp_ InternalKeyComparator Private Const		<i>Default:</i>
last_sequence_ uint64_t Private		<i>Default:</i>
log_number_ uint64_t Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
manifest_file_number_ uint64_t Private		<i>Default:</i>
next_file_number_ uint64_t Private		<i>Default:</i>
options_ Options* Private Const		<i>Default:</i>
prev_log_number_ uint64_t Private	0 or backing store for memtable being compacted	<i>Default:</i>
table_cache_ TableCache* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
AddLiveFiles() void Public		std::set<uint64_t>* [in] live
AppendVersion() void Private		Version* [in] v
ApproximateOffsetOf() uint64_t Public		Version* [in] v InternalKey& [in] key

Method	Notes	Parameters
CompactRange() Compaction* Public		int [in] level InternalKey* [in] begin InternalKey* [in] end
current() Version* Public		
EXCLUSIVE_LOCKS_R EQUIRED() Public		mu [in] prm1
Finalize() void Private		Version* [in] v
GetRange() void Private		std::vector<FileMetaData*>& [in] inputs InternalKey* [in] smallest InternalKey* [in] largest
GetRange2() void Private		std::vector<FileMetaData*>& [in] inputs1 std::vector<FileMetaData*>& [in] inputs2 InternalKey* [in] smallest InternalKey* [in] largest
LastSequence() uint64_t Public		
Const LevelSummary() char* Public		LevelSummaryStorage* [in] scratch
LogAndApply() Status Public		VersionEdit* [in] edit port::Mutex* [in] mu
LogNumber() uint64_t Public		
MakeInputIterator() Iterator* Public		Compaction* [in] c
ManifestFileNumber() uint64_t Public		

Method	Notes	Parameters
MarkFileNameUsed() void Public		uint64_t [in] number
MaxNextLevelOverlappingBytes() int64_t Public		
NeedsCompaction() bool Public		
NewFileName() uint64_t Public		
NumLevelBytes() int64_t Public		int [in] level
NumLevelFiles() int Public		int [in] level
operator=() void Private		VersionSet& [in] prm1
PickCompaction() Compaction* Public		
PrevLogNumber() uint64_t Public		
Recover() Status Public		
ReuseFileName() void Public		uint64_t [in] file_number
SetLastSequence() void Public		uint64_t [in] s
SetupOtherInputs() void Private		Compaction* [in] c
VersionSet() Public		std::string& [in] dbname Options* [in] options TableCache* [in] table_cache InternalKeyComparator* [in] prm1
~VersionSet() Public		
VersionSet() Private		VersionSet& [in] prm1
WriteSnapshot() Status Private		log::Writer* [in] log

Method	Notes	Parameters

LevelSummaryStorage

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {F2BDB3D6-EE48-4219-B2BC-5A9C4BBF5A48}

Return a human-readable short (single-line) summary of the number of files per level. Uses *scratch as backing store.

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
buffer char Public Collection		<i>Default:</i>

WritableFile

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: leveldb *Keywords:*

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {BDBA636D-F38C-490b-AE18-7583ED88EC83}

A file abstraction for sequential writing. The implementation must provide buffering since callers may append small fragments at a time to the file.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes

Connector	Source	Target	Notes
Association Source -> Destination	Public Writer	Private dest_ WritableFile	
Association Source -> Destination	Public DBImpl	Private logfile_ WritableFile	
Association Source -> Destination	Public VersionSet	Private descriptor_file_ WritableFile	

Operations

Method	Notes	Parameters
Pure Append() Status Public		Slice& [in] data
Pure Close() Status Public		
Pure Flush() Status Public		
operator=() void Private		WritableFile& [in] prm1
Pure Sync() Status Public		
WritableFile() Public		
~WritableFile() Public		
WritableFile() Private		WritableFile& [in] prm1

WriteBatch

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {2A1988BA-517D-4125-90CB-56A113206CD9}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association	Public	Private tmp_batch_	

Connector	Source	Target	Notes
Source -> Destination	DBImpl	WriteBatch	
Association Source -> Destination	Public CLevelDBBatch	Private batch WriteBatch	

Attributes

Attribute	Notes	Constraints and tags
rep_ std::string Private	See comment in write_batch.cc for the format of rep_-	<i>Default:</i>

Operations

Method	Notes	Parameters
Clear() void Public		
Delete() void Public		Slice& [in] key
Iterate() Status Public		Handler* [in] handler
Put() void Public		Slice& [in] key Slice& [in] value
WriteBatch() Public		
~WriteBatch() Public		

Handler

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {3D7C77E8-4BF1-496f-BB38-75828789B52D}

Support for iterating over the contents of a batch.

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
Pure Delete() void Public		Slice& [in] key
~Handler() Public		
Pure Put() void Public		Slice& [in] key Slice& [in] value

WriteBatchInternal**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** leveldb **Keywords:****Detail:** *Created on 8/30/2015. Last modified on 8/30/2015.***GUID:** {A056C12F-C8BB-499c-952C-097F3158CF32}

WriteBatchInternal provides static methods for manipulating a WriteBatch that we don't want in the public WriteBatch interface.

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
Static Append() void Public		WriteBatch* [in] dst WriteBatch* [in] src
Static ByteSize() size_t Public		WriteBatch* [in] batch
Static Contents() Slice Public		WriteBatch* [in] batch
Static Count() int Public		WriteBatch* [in] batch
Static InsertInto() Status Public		WriteBatch* [in] batch MemTable* [in] memtable

Method	Notes	Parameters
Static Sequence() SequenceNumber Public		WriteBatch* [in] batch
Static SetContents() void Public		WriteBatch* [in] batch Slice& [in] contents
Static SetCount() void Public		WriteBatch* [in] batch int [in] n
Static SetSequence() void Public		WriteBatch* [in] batch SequenceNumber [in] seq

WriteOptions

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {BBD0F9F1-BCF7-42ea-A5BD-A85537EBEAB1}

Options that control write operations

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CLevelDBWrapper	Private syncoptions WriteOptions	
Association Source -> Destination	Public CLevelDBWrapper	Private writeoptions WriteOptions	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
sync bool Public	If true, the write will be flushed from the operating system buffer cache (by calling WritableFile::Sync()) before the write is considered complete. If this flag is true, writes will be slower. If this flag is false, and the machine crashes, some recent writes may be lost. Note that if it is just the process that crashes (i.e., the machine does not reboot), no writes will be lost even if sync==false. In other words, a DB write with sync==false has similar crash semantics as the "write()" system call. A DB write with sync==true has similar crash semantics to a "write()" system call followed by "fsync()". Default: false	<i>Default:</i>

Operations

Method	Notes	Parameters
WriteOptions() Public		

log

Type: [Package](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb
Detail: *Created on 8/30/2015. Last modified on 8/30/2015*
GUID: {4F183E2A-72B1-4f46-85C8-0436446315F1}

log - (Logical diagram)

Created By: dev on 8/30/2015
Last Modified: 8/30/2015
Version: 1.0. [Locked:](#) False
GUID: {EA1502A1-2F2A-4cd5-BFE0-1DB906251F43}

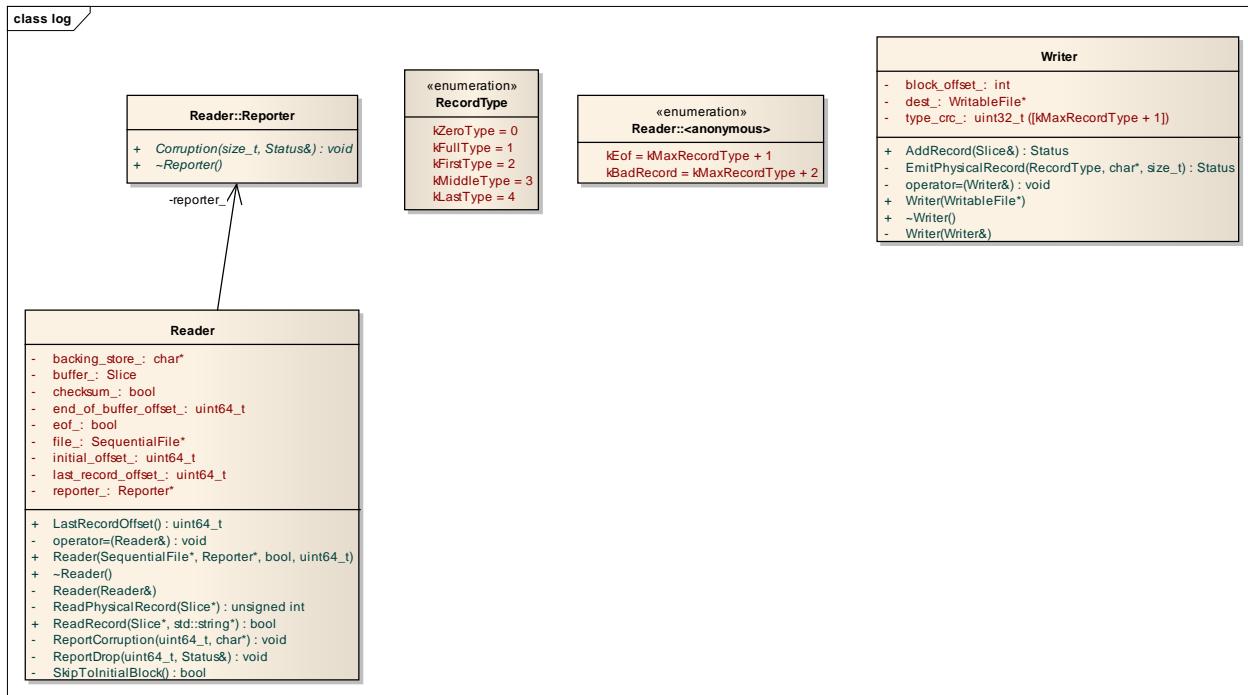


Figure: 4

Reader

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: log **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {15C32271-1BD4-43b4-92CD-249F411FC773}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Reader	Private reporter_ Reporter	
Association Source -> Destination	Public Reader	Private buffer_ Slice	
Association Source -> Destination	Public Reader	Private end_of_buffer_offset_ uint64_t	
Association	Public	Private file_	

Connector	Source	Target	Notes
Source -> Destination	Reader	SequentialFile	
Association Source -> Destination	Public Reader	Private last_record_offset_ uint64_t	
Association Source -> Destination	Public Reader	Private initial_offset_ uint64_t	

Attributes

Attribute	Notes	Constraints and tags
backing_store_ char* Private		<i>Default:</i>
buffer_ Slice Private		<i>Default:</i>
checksum_ bool Private		<i>Default:</i>
end_of_buffer_offset_ uint64_t Private	Offset of the first location past the end of buffer_.	<i>Default:</i>
eof_ bool Private	Last Read() indicated EOF by returning < kBlockSize	<i>Default:</i>

Attribute	Notes	Constraints and tags
file_ SequentialFile* Private		<i>Default:</i>
initial_offset_ uint64_t Private	Offset at which to start looking for the first record to return	<i>Default:</i>
last_record_offset_ uint64_t Private	Offset of the last record returned by ReadRecord.	<i>Default:</i>
reporter_ Reporter* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
LastRecordOffset() uint64_t Public		
operator=() void Private		Reader& [in] prm1
Reader() Public		SequentialFile* [in] file Reporter* [in] reporter bool [in] checksum uint64_t [in] initial_offset
~Reader() Public		

Method	Notes	Parameters
Reader() Private		Reader& [in] prm1
ReadPhysicalRecord() unsigned int Private		Slice* [in] result
ReadRecord() bool Public		Slice* [in] record std::string* [in] scratch
ReportCorruption() void Private		uint64_t [in] bytes char* [in] reason
ReportDrop() void Private		uint64_t [in] bytes Status& [in] reason
SkipToInitialBlock() bool Private		

<anonymous>

Type: **Enumeration**
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: log Keywords:
 Detail: Created on 8/30/2015. Last modified on 8/30/2015.
 GUID: {ABE678F3-F0EE-4136-9CBC-E36EF0B70C33}

Extend record types with the following special values

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
kEof Private		<i>Default:</i> kMaxRecordType + 1

Attribute	Notes	Constraints and tags
kBadRecord Private	Returned whenever we find an invalid physical record. Currently there are three situations in which this happens: The record has an invalid CRC (ReadPhysicalRecord reports a drop) The record is a 0-length record (No drop is reported) The record is below constructor's initial_offset (No drop is reported)	<i>Default:</i> kMaxRecordType + 2

Reporter

Type: [Class](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: log **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {9F5DA39F-BBBF-47a2-B4C1-77DDB9199978}

Interface for reporting errors.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Reader	Private reporter_ Reporter	

Operations

Method	Notes	Parameters
Pure Corruption() void Public		size_t [in] bytes Status& [in] status
~Reporter() Public		

RecordType

Type: [Enumeration](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: log **Keywords:**
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {F4DC7F26-42D5-4882-A990-345EB991D1D1}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
kZeroType Public	Zero is reserved for preallocated files	<i>Default:</i> 0
kFullType Public		<i>Default:</i> 1
kFirstType Public	For fragments	<i>Default:</i> 2
kMiddleType Public		<i>Default:</i> 3
kLastType Public		<i>Default:</i> 4

Writer

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: log
Keywords:
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {F7173EDD-41DB-4977-8FE1-B3760D994376}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Writer	Private dest_ WritableFile	
Association Source -> Destination	Public Writer	Private type_crc_ uint32_t	
Association Source -> Destination	Public DBImpl	Private log_ Writer	
Association Source -> Destination	Public VersionSet	Private descriptor_log_ Writer	

Attributes

Attribute	Notes	Constraints and tags
block_offset_ int Private	Current offset in block	<i>Default:</i>
dest_ WritableFile* Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
type_crc_ uint32_t Private Collection	crc32c values for all supported record types. These are pre-computed to reduce the overhead of computing the crc of the record type stored in the header.	<i>Default:</i>

Operations

Method	Notes	Parameters
AddRecord() Status Public		Slice& [in] slice
EmitPhysicalRecord() Status Private		RecordType [in] type char* [in] ptr size_t [in] length
operator=() void Private		Writer& [in] prm1
Writer() Public		WritableFile* [in] dest
~Writer() Public		
Writer() Private		Writer& [in] prm1

port

Type: [Package](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: leveldb
Detail: *Created on 8/30/2015. Last modified on 8/30/2015*
GUID: {7189C374-5972-422d-916D-D520E922E6DC}

port - (Logical diagram)

Created By: dev on 8/30/2015
Last Modified: 8/30/2015
Version: 1.0. [Locked](#): False
GUID: {765401B2-4BFB-4fa2-82E8-EFE27F8AF495}

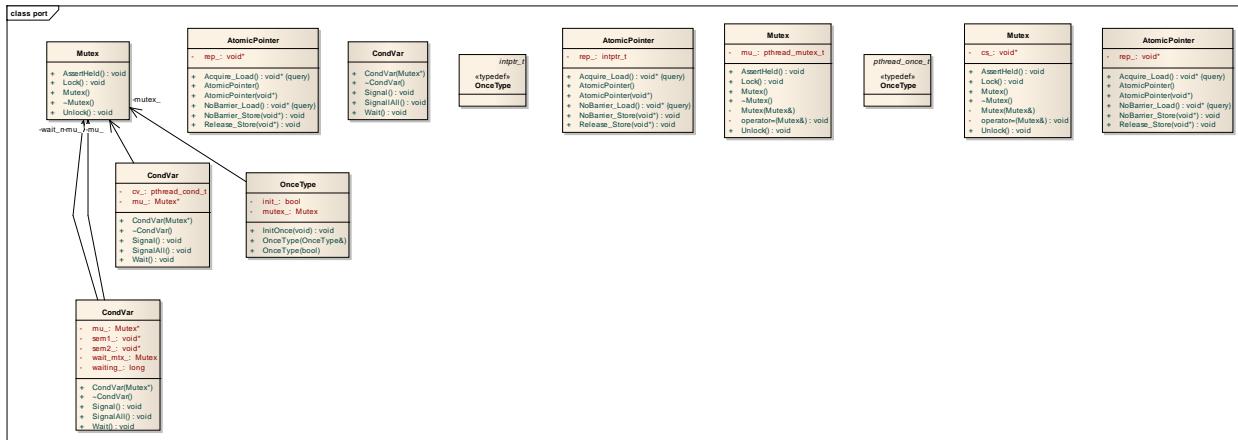


Figure: 5

AtomicPointer

Type:

Class

Status: Proposed. Version 1.0. Phase 1.0.

Package: port

Keywords:

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {EA80211E-2424-4a45-8EEB-E8145E2B4B16}

Storage for a lock-free pointer

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
rep_void*	Private	Default:

Operations

Method	Notes	Parameters
Acquire_Load() void*		
Public		
AtomicPointer()		
Public		
AtomicPointer()		void* [in] v
Public		

Method	Notes	Parameters
NoBarrier_Load() void*		
Public		
NoBarrier_Store() void		void* [in] v
Public		
Release_Store() void		void* [in] v
Public		

AtomicPointer

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: port **Keywords:**

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {71581B35-D015-4763-97F2-239CC5D16A69}

A type that holds a pointer that can be read or written atomically (i.e., without word-tearing.)

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
rep_ intptr_t Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Acquire_Load() void*		
Public		
AtomicPointer()		
Public		
AtomicPointer()		void* [in] v
Public		
NoBarrier_Load() void*		
Public		
NoBarrier_Store() void		void* [in] v
Public		

Method	Notes	Parameters
Release_Store() void Public		void* [in] v

AtomicPointer

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: port **Keywords:**

Detail: Created on 8/30/2015. Last modified on 8/30/2015.

GUID: {2D86D335-D630-4a2b-8944-6F819DDDB2A9}

AtomicPointer built using platform-specific MemoryBarrier()

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private shutting_down_ AtomicPointer	
Association Source -> Destination	Public SkipList	Private max_height_ AtomicPointer	
Association Source -> Destination	Public DBImpl	Private has_imm_ AtomicPointer	
Association Source -> Destination	Public Node	Private next_ AtomicPointer	

Attributes

Attribute	Notes	Constraints and tags
rep_void* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
Acquire_Load() void*		

Method	Notes	Parameters
Public		
AtomicPointer() Public		
AtomicPointer() Public		void* [in] p
NoBarrier_Load() void* Public		
NoBarrier_Store() void Public		void* [in] v
Release_Store() void Public		void* [in] v

CondVar

Type: **Class**

Status: Proposed. Version 1.0. Phase 1.0.

Package: port Keywords:

Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*

GUID: {7FB8914C-6736-4129-B2A8-36EDE75BEBD7}

the Win32 API offers a dependable condition variable mechanism, but only starting with Windows 2008 and Vista no matter what we will implement our own condition variable with a semaphore implementation as described in a paper written by Andrew D. Birrell in 2003

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public CondVar	Private mu_ Mutex	
Association Source -> Destination	Public CondVar	Private wait_mtx_ Mutex	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
mu_ Mutex*		<i>Default:</i>
Private		
sem1_ void*		<i>Default:</i>
Private		
sem2_ void*		<i>Default:</i>
Private		
wait_mtx_ Mutex		<i>Default:</i>
Private		
waiting_ long		<i>Default:</i>
Private		

Operations

Method	Notes	Parameters
CondVar()		Mutex* [in] mu
Public		
~CondVar()		
Public		
Signal()	void	
Public		
SignalAll()	void	
Public		

Method	Notes	Parameters
Wait() void Public		

CondVar

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: port **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {005EF17E-76DE-42dd-9B4F-3BF8EF027354}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
<u>Association</u> Source -> Destination	Public CondVar	Private mu_ Mutex	

Attributes

Attribute	Notes	Constraints and tags
cv_ pthread_cond_t Private		<i>Default:</i>
mu_ Mutex* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
CondVar() Public		Mutex* [in] mu

Method	Notes	Parameters
~CondVar() Public		
Signal() void Public		
SignalAll() void Public		
Wait() void Public		

CondVar

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: port **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {F580CB36-AD2E-48f7-A3A7-A99EEBC3044D}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private bg_cv_ CondVar	

Operations

Method	Notes	Parameters
CondVar() Public		Mutex* [in] mu
~CondVar() Public		
Signal() void Public		
SignalAll() void Public		
Wait() void Public		

Mutex

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.

Package: port *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {9952E73D-23C8-4e3c-9C59-E43F0446CECC}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
cs_void* Private	critical sections are more efficient than mutexes but they are not recursive and can only be used to synchronize threads within the same process we use opaque void * to avoid including windows.h in port_win.h	<i>Default:</i>

Operations

Method	Notes	Parameters
AssertHeld() void Public		
Lock() void Public		
Mutex() Public		
~Mutex() Public		
Mutex() Private		Mutex& [in] prm1
operator=() void Private		Mutex& [in] prm1
Unlock() void Public		

Mutex

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: port *Keywords:*
Detail: Created on 8/30/2015. Last modified on 8/30/2015.
GUID: {DA2FE683-62D3-4969-B362-B9D86AF14BA2}

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
mu_ pthread_mutex_t Private		<i>Default:</i>

Operations

Method	Notes	Parameters
AssertHeld() void Public		
Lock() void Public		
Mutex() Public		
~Mutex() Public		
Mutex() Private		Mutex& [in] prm1
operator=() void Private		Mutex& [in] prm1
Unlock() void Public		

Mutex

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: port **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {E5AF8300-9E3C-4a51-AADC-1B197196D9EE}

A Mutex represents an exclusive lock.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public DBImpl	Private mutex_ Mutex	
Association Source -> Destination	Public CondVar	Private mu_ Mutex	
Association Source -> Destination	Public CondVar	Private mu_ Mutex	
Association Source -> Destination	Public CondVar	Private wait_mtx_ Mutex	
Association Source -> Destination	Public OnceType	Private mutex_ Mutex	

Operations

Method	Notes	Parameters
AssertHeld() void Public		
Lock() void Public		
Mutex() Public		
~Mutex() Public		
Unlock() void Public		

OnceType

Type: Class
Status: Proposed. Version 1.0. Phase 1.0.
Package: port **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {D4872F2F-9327-40c1-A48A-FE8FF65EE9D}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public OnceType	Private mutex_ Mutex	

Connector	Source	Target	Notes

Attributes

Attribute	Notes	Constraints and tags
init_ bool Private		<i>Default:</i>
mutex_ Mutex Private		<i>Default:</i>

Operations

Method	Notes	Parameters
InitOnce() void Public		void [in] prm1
OnceType() Public		OnceType& [in] once
OnceType() Public		bool [in] f

OnceType

Type: Class **pthread_once_t**
Status: Proposed. Version 1.0. Phase 1.0.
Package: port **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {F8407A3E-A3DF-44ed-A477-EFE96CA5CD22}

Custom Properties

- isActive = False

OnceType

Type:	Class	intptr_t
Status:	Proposed.	Version 1.0. Phase 1.0.
Package:	port	Keywords:
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>	
GUID:	{30D8EEEE-B08A-4a47-ABE5-993CB20C8B52}	

Thread-safe initialization. Used as follows: static port::OnceType init_control = LEVELDB_ONCE_INIT; static void Initializer() { ... do something ...; } ... port::InitOnce(&init_control, &Initializer);

Custom Properties

- isActive = False

test

Type:	Package
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	leveldb
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015</i>
GUID:	{84B44126-CB22-49f2-B6F9-BF1DCE2E5CEE}

test - (Logical diagram)

Created By:	dev	on 8/30/2015
Last Modified:	8/30/2015	
Version:	1.0.	Locked: False
GUID:	{DF62E93A-4C3B-4668-970F-43F8B49B6806}	

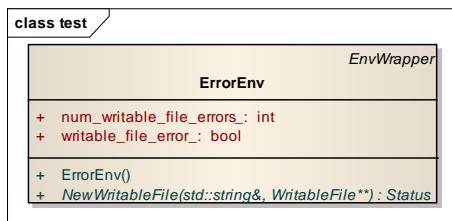


Figure: 6

ErrorEnv

Type:	Class	EnvWrapper
Status:	Proposed.	Version 1.0. Phase 1.0.
Package:	test	Keywords:
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015.</i>	
GUID:	{322B5059-EEE8-4871-ABFC-D91D06E97B48}	

A wrapper that allows injection of errors.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public ErrorEnv	Public EnvWrapper	

Attributes

Attribute	Notes	Constraints and tags
num_writable_file_errors _ int Public		<i>Default:</i>
writable_file_error_ bool Public		<i>Default:</i>

Operations

Method	Notes	Parameters
ErrorEnv() Public		
NewWritableFile() Status Public		std::string& [in] fname WritableFile** [in] result

memusage

Type:	Package
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015</i>
GUID:	{D36A2813-B6B2-498c-AF9A-4B7199B3F7E0}

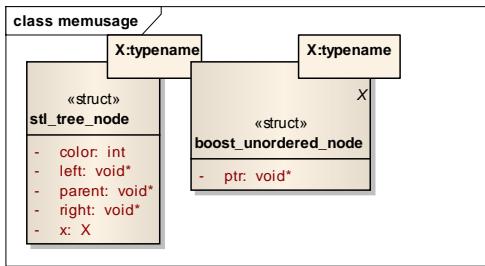
memusage - (Logical diagram)**Created By:** dev on 8/30/2015**Last Modified:** 8/30/2015**Version:** 1.0. **Locked:** False**GUID:** {5FAA1178-5725-4985-B968-442C9D195F26}

Figure: 7

boost_unordered_node**Type:** Class X**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** memusage **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {056D6CF6-EAF5-45a8-8D9E-E875FD33930D}

Boost data structures

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
ptr void* Private		<i>Default:</i>

stl_tree_node**Type:** Class**Status:** Proposed. Version 1.0. Phase 1.0.**Package:** memusage **Keywords:****Detail:** Created on 8/30/2015. Last modified on 8/30/2015.**GUID:** {252B9C0D-7FF8-4592-894F-495FC6A6C2AB}

STL data structures

Custom Properties

- isActive = False

Attributes

Attribute	Notes	Constraints and tags
color int Private		<i>Default:</i>
left void* Private		<i>Default:</i>
parent void* Private		<i>Default:</i>
right void* Private		<i>Default:</i>
x X Private		<i>Default:</i>

payments

Type:	<u>Package</u>
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model
Detail:	<i>Created on 8/30/2015. Last modified on 8/30/2015</i>
GUID:	{3A3DEF47-B72D-4391-86FF-0BB1B475C86E}

payments - (*Logical diagram*)

Created By: dev on 8/30/2015
Last Modified: 8/30/2015
Version: 1.0. *Locked:* False
GUID: {A96AB6F6-0F7B-4f82-A3FD-B2526931EE30}



Figure: 8

Output

Type: `Class ::google::protobuf::Message`
Status: Proposed. Version 1.0. Phase 1.0.
Package: payments **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {7D356069-08E8-4a91-A8AA-6CA25E794F3B}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Output	Private default_instance_ Output	

Attributes

Attribute	Notes	Constraints and tags
<code>_cached_size_</code> int Private		<i>Default:</i> <code>[mutable = true]</code>
<code>_has_bits_</code> <code>::google::protobuf::uint32</code> Private Collection		<i>Default:</i>
<code>_unknown_fields_</code> <code>::google::protobuf::UnknownFieldSet</code> Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
amount_ ::google::protobuf::uint64 Private		<i>Default:</i>
default_instance_ Output* Private Static		<i>Default:</i>
kAmountFieldNumber int Public Static Const		<i>Default:</i> 1
kScriptFieldNumber int Public Static Const		<i>Default:</i> 2
script_ ::std::string* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
amount() ::google::protobuf::uint64 Public		
ByteSize() int Public		
Clear() void Public		
clear_amount() void Public		

Method	Notes	Parameters
clear_has_amount() void Private		
clear_has_script() void Private		
clear_script() void Public		
CopyFrom() void Public		::google::protobuf::Message& [in] from
CopyFrom() void Public		Output& [in] from
Static Const default_instance() Output& Public		
Static Const descriptor() ::google::protobuf::Descriptor* Public		
GetCachedSize() int Public		
GetMetadata() ::google::protobuf::Metadata Public		
has_amount() bool Public		
has_script() bool Public		
InitAsDefaultInstance() void Private		
IsInitialized() bool Public		
MergeFrom() void Public		::google::protobuf::Message& [in] from
MergeFrom() void Public		Output& [in] from
MergePartialFromCodedStream() bool Public		::google::protobuf::io::CodedInputStream* [in] input
mutable_script() ::std::string* Public		
mutable_unknown_fields 0 ::google::protobuf::UnknownFieldSet*		

Method	Notes	Parameters
Public		
New() Output*		
Public		
operator=() Output&		Output& [in] from
Public		
Output()		
Public		
~Output()		
Public		
Output()		Output& [in] from
Public		
protobuf_AddDesc_payoutrequest_2eprot() void		
Private		
protobuf_AssignDesc_paymentrequest_2eprot() void		
Private		
protobuf_ShutdownFile_paymentrequest_2eprot() void		
Private		
release_script()		
<code>::std::string*</code>		
Public		
Const script()		
<code>::std::string&</code>		
Public		
SerializeWithCachedSize_s() void		<code>::google::protobuf::io::CodedOutputStream*</code> [in] output
Public		
SerializeWithCachedSize_sToArray()		<code>::google::protobuf::uint8*</code> [in] output
<code>::google::protobuf::uint8*</code>		
Public		
set_allocated_script()		<code>::std::string*</code> [in] script
void		
Public		
set_amount() void		<code>::google::protobuf::uint64</code> [in] value
Public		
set_has_amount() void		
Private		
set_has_script() void		
Private		
set_script() void		<code>::std::string&</code> [in] value
Public		
set_script() void		<code>char*</code> [in] value
Public		

Method	Notes	Parameters
set_script() void Public		void* [in] value size_t [in] size
SetCachedSize() void Private		int [in] size
SharedCtor() void Private		
SharedDtor() void Private		
Swap() void Public		Output* [in] other
Const unknown_fields() ::google::protobuf::UnknownFieldSet& Public		

Payment

Type: [**Class**](#) [**::google::protobuf::Message**](#)
Status: Proposed. Version 1.0. Phase 1.0.
Package: payments **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {7358B7D6-A508-4e01-809C-D8D08B5C97C6}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Payment	Private default_instance_Payment	
Association Source -> Destination	Public PaymentACK	Private payment_Payment	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
<code>_cached_size_ int</code> Private		<i>Default:</i> <code>[mutable = true]</code>
<code>_has_bits_</code> <code>::google::protobuf::uint32</code> Private Collection		<i>Default:</i>
<code>_unknown_fields_</code> <code>::google::protobuf::UnknownFieldSet</code> Private		<i>Default:</i>
<code>default_instance_</code> <code>Payment*</code> Private Static		<i>Default:</i>
<code>kMemoFieldNumber int</code> Public Static Const		<i>Default:</i> 4
<code>kMerchantDataFieldNumber int</code> Public Static Const		<i>Default:</i> 1

Attribute	Notes	Constraints and tags
kRefundToFieldNumber int Public Static Const		<i>Default:</i> 3
kTransactionsFieldNumber int Public Static Const		<i>Default:</i> 2
memo_ ::std::string* Private		<i>Default:</i>
merchant_data_ ::std::string* Private		<i>Default:</i>
refund_to_ ::google::protobuf::RepeatingPtrField< ::payments::Output > Private		<i>Default:</i>
transactions_ ::google::protobuf::RepeatingPtrField< ::std::string> Private		<i>Default:</i>

Operations

Method	Notes	Parameters
add_refund_to() ::payments::Output* Public		
add_transactions() ::std::string* Public		
add_transactions() void Public		::std::string& [in] value
add_transactions() void Public		char* [in] value
add_transactions() void Public		void* [in] value size_t [in] size
ByteSize() int Public		
Clear() void Public		
clear_has_memo() void Private		
clear_has_merchant_data() () void Private		
clear_memo() void Public		
clear_merchant_data() void Public		
clear_refund_to() void Public		
clear_transactions() void Public		
CopyFrom() void Public		::google::protobuf::Message& [in] from
CopyFrom() void Public		Payment& [in] from
Static Const default_instance() Payment& Public		
Static Const descriptor() ::google::protobuf::Descriptor* Public		
GetCachedSize() int Public		
GetMetadata()		

Method	Notes	Parameters
::google::protobuf::Metadata Public		
has_memo() bool Public		
has_merchant_data() bool Public		
InitAsDefaultInstance() void Private		
IsInitialized() bool Public		
Const memo() ::std::string& Public		
Const merchant_data() ::std::string& Public		
MergeFrom() void Public		::google::protobuf::Message& [in] from
MergeFrom() void Public		Payment& [in] from
MergePartialFromCodedStream() bool Public		::google::protobuf::io::CodedInputStream* [in] input
mutable_memo() ::std::string* Public		
mutable_merchant_data() ::std::string* Public		
mutable_refund_to() ::payments::Output* Public		int [in] index
mutable_refund_to() ::google::protobuf::RepeatingPtrField< ::payments::Output >* Public		
mutable_transactions() ::std::string* Public		int [in] index
mutable_transactions() ::google::protobuf::RepeatingPtrField< ::std::string>* Public		
mutable_unknown_fields 0 ::google::protobuf::UnknownFields		

Method	Notes	Parameters
wnFieldSet*		
Public		
New() Payment*		
Public		
operator=() Payment&		Payment& [in] from
Public		
Payment()		
Public		
~Payment()		
Public		
Payment()		Payment& [in] from
Public		
protobuf_AddDesc_paym entrequest_2eproto() void		
Private		
protobuf_AssignDesc_pa ymentrequest_2eproto() void		
Private		
protobuf_ShutdownFile_ paymentrequest_2eproto() void		
Private		
Const refund_to() ::payments::Output&		int [in] index
Public		
Const refund_to() ::google::protobuf::Repeate dPtrField< ::payments::Output >&		
Public		
refund_to_size() int		
Public		
release_memo()		
::std::string*		
Public		
release_merchant_data()		
::std::string*		
Public		
SerializeWithCachedSize s() void		::google::protobuf::io::CodedOut putStream* [in] output
Public		
SerializeWithCachedSize sToArray()		::google::protobuf::uint8* [in]
::google::protobuf::uint8*		output
Public		
set_allocated_memo()		::std::string* [in] memo
void		
Public		
set_allocated_merchant_ data() void		::std::string* [in] merchant_data

Method	Notes	Parameters
Public		
set_has_memo() void		
Private		
set_has_merchant_data() void		
Private		
set_memo() void		::std::string& [in] value
Public		
set_memo() void		char* [in] value
Public		
set_memo() void		char* [in] value
Public		size_t [in] size
set_merchant_data() void		::std::string& [in] value
Public		
set_merchant_data() void		char* [in] value
Public		
set_merchant_data() void		void* [in] value
Public		size_t [in] size
set_transactions() void		int [in] index
Public		::std::string& [in] value
set_transactions() void		int [in] index
Public		char* [in] value
set_transactions() void		int [in] index
Public		void* [in] value
		size_t [in] size
SetCachedSize() void		int [in] size
Private		
SharedCtor() void		
Private		
SharedDtor() void		
Private		
Swap() void		Payment* [in] other
Public		

Method	Notes	Parameters
Const transactions() ::std::string& Public		int [in] index
Const transactions() ::google::protobuf::RepeatingPtrField< ::std::string>& Public		
transactions_size() int Public		
Const unknown_fields() ::google::protobuf::UnknownFieldSet& Public		

PaymentACK

Type: Class **::google::protobuf::Message**
Status: Proposed. Version 1.0. Phase 1.0.
Package: payments **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {B9B19A1F-CB90-4f35-9C56-448AC255B627}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public PaymentACK	Private default_instance_ PaymentACK	
Association Source -> Destination	Public PaymentACK	Private payment_ Payment	

Attributes

Attribute	Notes	Constraints and tags

Attribute	Notes	Constraints and tags
<code>_cached_size_ int</code> Private		<i>Default:</i> <code>[mutable = true]</code>
<code>_has_bits_</code> <code>::google::protobuf::uint32</code> Private Collection		<i>Default:</i>
<code>_unknown_fields_</code> <code>::google::protobuf::UnknownFieldSet</code> Private		<i>Default:</i>
<code>default_instance_</code> <code>PaymentACK*</code> Private Static		<i>Default:</i>
<code>kMemoFieldNumber</code> int Public Static Const		<i>Default:</i> 2
<code>kPaymentFieldNumber</code> int Public Static Const		<i>Default:</i> 1

Attribute	Notes	Constraints and tags
memo_ ::std::string* Private		<i>Default:</i>
payment_ ::payments::Payment* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
ByteSize() int Public		
Clear() void Public		
clear_has_memo() void Private		
clear_has_payment() void Private		
clear_memo() void Public		
clear_payment() void Public		
CopyFrom() void Public		::google::protobuf::Message& [in] from
CopyFrom() void Public		PaymentACK& [in] from
Static Const default_instance() PaymentACK& Public		
Static Const descriptor() ::google::protobuf::Descriptor* Public		
GetCachedSize() int Public		
GetMetadata() ::google::protobuf::Metadata Public		
has_memo() bool		

Method	Notes	Parameters
Public		
has_payment() bool		
Public		
InitAsDefaultInstance()		
void		
Private		
IsInitialized() bool		
Public		
Const memo()		
::std::string&		
Public		
MergeFrom() void		
Public		::google::protobuf::Message& [in] from
MergeFrom() void		
Public		PaymentACK& [in] from
MergePartialFromCodedStream() bool		
Public		::google::protobuf::io::CodedInputStream* [in] input
mutable_memo()		
::std::string*		
Public		
mutable_payment()		
::payments::Payment*		
Public		
mutable_unknown_fields		
0		
::google::protobuf::UnknownFieldSet*		
Public		
New() PaymentACK*		
Public		
operator=()		
PaymentACK&		
Public		PaymentACK& [in] from
Const payment()		
::payments::Payment&		
Public		
PaymentACK()		
Public		
~PaymentACK()		
Public		
PaymentACK()		
Public		PaymentACK& [in] from
protobuf_AddDesc_paymentrequest_2eproto() void		
Private		
protobuf_AssignDesc_paymentrequest_2eproto()		
void		

Method	Notes	Parameters
Private		
protobuf_ShutdownFile_paymentrequest_2eproto() void		
Private		
release_memo() ::std::string*		
Public		
release_payment() ::payments::Payment*		
Public		
SerializeWithCachedSize() void		::google::protobuf::io::CodedOutputStream* [in] output
Public		
SerializeWithCachedSizeToArray() ::google::protobuf::uint8*		::google::protobuf::uint8* [in] output
Public		
set_allocated_memo() void		::std::string* [in] memo
Public		
set_allocated_payment() void		::payments::Payment* [in] payment
Public		
set_has_memo() void		
Private		
set_has_payment() void		
Private		
set_memo() void		::std::string& [in] value
Public		
set_memo() void		char* [in] value
Public		
set_memo() void		char* [in] value size_t [in] size
Public		
SetCachedSize() void		int [in] size
Private		
SharedCtor() void		
Private		
SharedDtor() void		
Private		
Swap() void		PaymentACK* [in] other
Public		
Const unknown_fields() ::google::protobuf::UnknownFieldSet&		

Method	Notes	Parameters
Public		

PaymentDetails

Type: Class **::google::protobuf::Message**
Status: Proposed. Version 1.0. Phase 1.0.
Package: payments **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {69DC8718-56EF-4cce-8E6F-B42FD9C65BF2}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public PaymentDetails	Private default_instance_ PaymentDetails	
Association Source -> Destination	Public PaymentRequestPlus	Private details PaymentDetails	

Attributes

Attribute	Notes	Constraints and tags
_cached_size_ int Private		<i>Default:</i> [<u>mutable</u> = true]
_default_network_ ::std::string* Private Static		<i>Default:</i>

Attribute	Notes	Constraints and tags
<code>_has_bits_</code> ::google::protobuf::uint32 Private Collection		<i>Default:</i>
<code>_unknown_fields_</code> ::google::protobuf::UnknownFieldSet Private		<i>Default:</i>
<code>default_instance_</code> PaymentDetails* Private Static		<i>Default:</i>
<code>expires_</code> ::google::protobuf::uint64 Private		<i>Default:</i>
<code>kExpiresFieldNumber</code> int Public Static Const		<i>Default:</i> 4
<code>kMemoFieldNumber</code> int Public Static Const		<i>Default:</i> 5

Attribute	Notes	Constraints and tags
kMerchantDataFieldNumber int Public Static Const		<i>Default:</i> 7
kNetworkFieldNumber int Public Static Const		<i>Default:</i> 1
kOutputsFieldNumber int Public Static Const		<i>Default:</i> 2
kPaymentUrlFieldNumber int Public Static Const		<i>Default:</i> 6
kTimeFieldNumber int Public Static Const		<i>Default:</i> 3
memo_ ::std::string* Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
merchant_data_ ::std::string* Private		<i>Default:</i>
network_ ::std::string* Private		<i>Default:</i>
outputs_ ::google::protobuf::RepeatingPtrField< ::payments::Output > Private		<i>Default:</i>
payment_url_ ::std::string* Private		<i>Default:</i>
time_ ::google::protobuf::uint64 Private		<i>Default:</i>

Operations

Method	Notes	Parameters
add_outputs() ::payments::Output* Public		
ByteSize() int Public		
Clear() void Public		

Method	Notes	Parameters
clear_expires() void Public		
clear_has_expires() void Private		
clear_has_memo() void Private		
clear_has_merchant_data() void Private		
clear_has_network() void Private		
clear_has_payment_url() void Private		
clear_has_time() void Private		
clear_memo() void Public		
clear_merchant_data() void Public		
clear_network() void Public		
clear_outputs() void Public		
clear_payment_url() void Public		
clear_time() void Public		
CopyFrom() void Public		::google::protobuf::Message& [in] from
CopyFrom() void Public		PaymentDetails& [in] from
Static Const default_instance() PaymentDetails& Public		
Static Const descriptor() ::google::protobuf::Descriptor* Public		
expires() ::google::protobuf::uint64 Public		
GetCachedSize() int Public		
GetMetadata() ::google::protobuf::Metadata Public		
has_expires() bool		

Method	Notes	Parameters
Public		
has_memo() bool		
Public		
has_merchant_data() bool		
Public		
has_network() bool		
Public		
has_payment_url() bool		
Public		
has_time() bool		
Public		
InitAsDefaultInstance() void		
Private		
IsInitialized() bool		
Public		
Const memo() ::std::string&		
Public		
Const merchant_data() ::std::string&		
Public		
MergeFrom() void		::google::protobuf::Message& [in]
Public		from
MergeFrom() void		PaymentDetails& [in] from
Public		
MergePartialFromCodedStream() bool		::google::protobuf::io::CodedInputStream* [in] input
Public		
mutable_memo() ::std::string*		
Public		
mutable_merchant_data() ::std::string*		
Public		
mutable_network() ::std::string*		
Public		
mutable_outputs() ::payments::Output*		int [in] index
Public		
mutable_outputs() ::google::protobuf::RepeatingPtrField<		
::payments::Output >*		
Public		
mutable_payment_url() ::std::string*		
Public		

Method	Notes	Parameters
mutable_unknown_fields() ::google::protobuf::UnknownFieldSet* Public		
Const network() ::std::string& Public		
New() PaymentDetails* Public		
operator=(PaymentDetails& Public		PaymentDetails& [in] from
Const outputs() ::payments::Output& Public		int [in] index
Const outputs() ::google::protobuf::RepeatedPtrField< ::payments::Output >& Public		
outputs_size() int Public		
Const payment_url() ::std::string& Public		
PaymentDetails() Public		
~PaymentDetails() Public		
PaymentDetails() Public		PaymentDetails& [in] from
protobuf_AddDesc_paymentrequest_2eprotocol () void Private		
protobuf_AssignDesc_paymentrequest_2eprotocol () void Private		
protobuf_ShutdownFile_paymentrequest_2eprotocol () void Private		
release_memo() ::std::string* Public		
release_merchant_data() ::std::string* Public		
release_network() ::std::string* Public		
release_payment_url()		

Method	Notes	Parameters
::std::string* Public		
SerializeWithCachedSize s() void Public		::google::protobuf::io::CodedOutputStream* [in] output
SerializeWithCachedSize sToArray() ::google::protobuf::uint8* Public		::google::protobuf::uint8* [in] output
set_allocated_memo() void Public		::std::string* [in] memo
set_allocated_merchant_data() void Public		::std::string* [in] merchant_data
set_allocated_network() void Public		::std::string* [in] network
set_allocated_payment_url() void Public		::std::string* [in] payment_url
set_expires() void Public		::google::protobuf::uint64 [in] value
set_has_expires() void Private		
set_has_memo() void Private		
set_has_merchant_data() void Private		
set_has_network() void Private		
set_has_payment_url() void Private		
set_has_time() void Private		
set_memo() void Public		::std::string& [in] value
set_memo() void Public		char* [in] value
set_memo() void Public		char* [in] value size_t [in] size
set_merchant_data() void Public		::std::string& [in] value

Method	Notes	Parameters
set_merchant_data() void Public		char* [in] value
set_merchant_data() void Public		void* [in] value size_t [in] size
set_network() void Public		::std::string& [in] value
set_network() void Public		char* [in] value
set_network() void Public		char* [in] value size_t [in] size
set_payment_url() void Public		::std::string& [in] value
set_payment_url() void Public		char* [in] value
set_payment_url() void Public		char* [in] value size_t [in] size
set_time() void Public		::google::protobuf::uint64 [in] value
SetCachedSize() void Private		int [in] size
SharedCtor() void Private		
SharedDtor() void Private		
Swap() void Public		PaymentDetails* [in] other
time() ::google::protobuf::uint64 Public		
Const unknown_fields() ::google::protobuf::UnknownFieldSet& Public		

PaymentRequest

Type: Class **::google::protobuf::Message**
Status: Proposed. Version 1.0. Phase 1.0.
Package: payments **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {C906E5A2-B7B2-48f2-9D0B-5CC4CCD80729}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public PaymentRequest	Private default_instance_ PaymentRequest	
Association Source -> Destination	Public PaymentRequestPlus	Private paymentRequest PaymentRequest	

Attributes

Attribute	Notes	Constraints and tags
<code>_cached_size_</code> int Private		<i>Default:</i> [<u>mutable</u> = true]
<code>_default_pki_type_</code> ::std::string* Private Static		<i>Default:</i>
<code>_has_bits_</code> ::google::protobuf::uint32 Private Collection		<i>Default:</i>

Attribute	Notes	Constraints and tags
<code>_unknown_fields_</code> ::google::protobuf::UnknownFieldSet Private		<i>Default:</i>
<code>default_instance_</code> PaymentRequest* Private Static		<i>Default:</i>
kPaymentDetailsVersion FieldNumber int Public Static Const		<i>Default:</i> 1
kPkiDataFieldNumber int Public Static Const		<i>Default:</i> 3
kPkiTypeFieldNumber int Public Static Const		<i>Default:</i> 2
kSerializedPaymentDetailFieldNumber int Public Static Const		<i>Default:</i> 4

Attribute	Notes	Constraints and tags
kSignatureFieldNumber int Public Static Const		<i>Default:</i> 5
payment_details_version — ::google::protobuf::uint32 Private		<i>Default:</i>
pki_data_ ::std::string* Private		<i>Default:</i>
pki_type_ ::std::string* Private		<i>Default:</i>
serialized_payment_detai ls_ ::std::string* Private		<i>Default:</i>
signature_ ::std::string* Private		<i>Default:</i>

Operations

Method	Notes	Parameters
--------	-------	------------

Method	Notes	Parameters
ByteSize() int Public		
Clear() void Public		
clear_has_payment_detail() Is_version() void Private		
clear_has_pki_data() void Private		
clear_has_pki_type() void Private		
clear_has_serialized_payment_details() void Private		
clear_has_signature() void Private		
clear_payment_details_version() void Public		
clear_pki_data() void Public		
clear_pki_type() void Public		
clear_serialized_payment_details() void Public		
clear_signature() void Public		
CopyFrom() void Public		<code>::google::protobuf::Message&</code> [in] from
CopyFrom() void Public		<code>PaymentRequest&</code> [in] from
Static Const default_instance() PaymentRequest& Public		
Static Const descriptor() <code>::google::protobuf::Descriptor*</code> Public		
GetCachedSize() int Public		
GetMetadata() <code>::google::protobuf::Metadata</code> Public		
has_payment_details_version() bool Public		
has_pki_data() bool		

Method	Notes	Parameters
Public		
has_pki_type() bool		
Public		
has_serialized_payment_details() bool		
Public		
has_signature() bool		
Public		
InitAsDefaultInstance()		
void		
Private		
IsInitialized() bool		
Public		
MergeFrom() void		::google::protobuf::Message& [in] from
Public		
MergeFrom() void		PaymentRequest& [in] from
Public		
MergePartialFromCodedStream() bool		::google::protobuf::io::CodedInputStream* [in] input
Public		
mutable_pki_data()		
::std::string*		
Public		
mutable_pki_type()		
::std::string*		
Public		
mutable_serialized_payment_details() ::std::string*		
Public		
mutable_signature()		
::std::string*		
Public		
mutable_unknown_fields		
0		
::google::protobuf::UnknownFieldSet*		
Public		
New() PaymentRequest*		
Public		
operator=()		PaymentRequest& [in] from
PaymentRequest&		
Public		
payment_details_version()		
::google::protobuf::uint32		
Public		
PaymentRequest()		
Public		
~PaymentRequest()		
Public		
PaymentRequest()		PaymentRequest& [in] from

Method	Notes	Parameters
Public		
Const pki_data() ::std::string& Public		
Const pki_type() ::std::string& Public		
protobuf_AddDesc_paym entrequest_2eproto() void Private		
protobuf_AssignDesc_pa ymentrequest_2eproto() void Private		
protobuf_ShutdownFile_ paymentrequest_2eproto() void Private		
release_pki_data() ::std::string* Public		
release_pki_type() ::std::string* Public		
release_serialized_payme nt_details() ::std::string* Public		
release_signature() ::std::string* Public		
Const serialized_payment_detai ls() ::std::string& Public		
SerializeWithCachedSize s() void Public		::google::protobuf::io::CodedOut putStream* [in] output
SerializeWithCachedSize sToArray() ::google::protobuf::uint8* Public		::google::protobuf::uint8* [in] output
set_allocated_pki_data() void Public		::std::string* [in] pki_data
set_allocated_pki_type() void Public		::std::string* [in] pki_type
set_allocated_serialized_ payment_details() void Public		::std::string* [in] serialized_payment_details
set_allocated_signature()		::std::string* [in] signature

Method	Notes	Parameters
void Public set_has_payment_details _version() void Private		
set_has_pki_data() void Private		
set_has_pki_type() void Private		
set_has_serialized_payme nt_details() void Private		
set_has_signature() void Private		
set_payment_details_verse ion() void Public		<code>::google::protobuf::uint32</code> [in] value
set_pki_data() void Public		<code>::std::string&</code> [in] value
set_pki_data() void Public		<code>char*</code> [in] value
set_pki_data() void Public		<code>void*</code> [in] value <code>size_t</code> [in] size
set_pki_type() void Public		<code>::std::string&</code> [in] value
set_pki_type() void Public		<code>char*</code> [in] value
set_pki_type() void Public		<code>char*</code> [in] value <code>size_t</code> [in] size
set_serialized_payment_d etails() void Public		<code>::std::string&</code> [in] value
set_serialized_payment_d etails() void Public		<code>char*</code> [in] value
set_serialized_payment_d etails() void Public		<code>void*</code> [in] value <code>size_t</code> [in] size
set_signature() void Public		<code>::std::string&</code> [in] value

Method	Notes	Parameters
set_signature() void Public		char* [in] value
set_signature() void Public		void* [in] value size_t [in] size
SetCachedSize() void Private		int [in] size
SharedCtor() void Private		
SharedDtor() void Private		
Const signature() ::std::string& Public		
Swap() void Public		PaymentRequest* [in] other
Const unknown_fields() ::google::protobuf::UnknownFieldSet& Public		

X509Certificates

Type: Class **::google::protobuf::Message**
Status: Proposed. Version 1.0. Phase 1.0.
Package: payments **Keywords:**
Detail: *Created on 8/30/2015. Last modified on 8/30/2015.*
GUID: {ECF2328A-5B7D-481e-A346-2C857560FCEB}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public X509Certificates	Private default_instance_X509Certificates	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
<code>_cached_size_</code> int Private		<i>Default:</i> <code>[mutable = true]</code>
<code>_has_bits_</code> ::google::protobuf::uint32 Private Collection		<i>Default:</i>
<code>_unknown_fields_</code> ::google::protobuf::UnknownFieldSet Private		<i>Default:</i>
<code>certificate_</code> ::google::protobuf::RepeatedPtrField< ::std::string> Private		<i>Default:</i>
<code>default_instance_</code> X509Certificates* Private Static		<i>Default:</i>
<code>kCertificateFieldNumber</code> int Public Static Const		<i>Default:</i> 1

Operations

Method	Notes	Parameters
add_certificate() ::std::string* Public		
add_certificate() void Public		::std::string& [in] value
add_certificate() void Public		char* [in] value
add_certificate() void Public		void* [in] value size_t [in] size
ByteSize() int Public		
Const certificate() ::std::string& Public		int [in] index
Const certificate() ::google::protobuf::RepeatingPtrField< ::std::string>& Public		
certificate_size() int Public		
Clear() void Public		
clear_certificate() void Public		
CopyFrom() void Public		::google::protobuf::Message& [in] from
CopyFrom() void Public		X509Certificates& [in] from
Static Const default_instance() X509Certificates& Public		
Static Const descriptor() ::google::protobuf::Descriptor* Public		
GetCachedSize() int Public		
GetMetadata() ::google::protobuf::Metadata Public		
InitAsDefaultInstance() void Private		
IsInitialized() bool		

Method	Notes	Parameters
Public		
MergeFrom() void Public		::google::protobuf::Message& [in] from
MergeFrom() void Public		X509Certificates& [in] from
MergePartialFromCodedStream() bool Public		::google::protobuf::io::CodedInputStream* [in] input
mutable_certificate() ::std::string* Public		int [in] index
mutable_certificate() ::google::protobuf::RepeatedPtrField< ::std::string>* Public		
mutable_unknown_fields 0 ::google::protobuf::UnknownFieldSet* Public		
New() X509Certificates* Public		
operator=() X509Certificates& Public		X509Certificates& [in] from
protobuf_AddDesc_paymentrequest_2eprotocol() void Private		
protobuf_AssignDesc_paymentrequest_2eprotocol() void Private		
protobuf_ShutdownFile_paymentrequest_2eprotocol() void Private		
SerializeWithCachedSizes() void Public		::google::protobuf::io::CodedOutputStream* [in] output
SerializeWithCachedSizesToArray() ::google::protobuf::uint8* Public		::google::protobuf::uint8* [in] output
set_certificate() void Public		int [in] index ::std::string& [in] value
set_certificate() void		int [in] index

Method	Notes	Parameters
Public		char* [in] value
set_certificate() void Public		int [in] index void* [in] value size_t [in] size
SetCachedSize() void Private		int [in] size
SharedCtor() void Private		
SharedDtor() void Private		
Swap() void Public		X509Certificates* [in] other
Const unknown_fields() ::google::protobuf::UnknownFieldSet& Public		
X509Certificates() Public		
~X509Certificates() Public		
X509Certificates() Public		X509Certificates& [in] from