

Keywords	Commit ID	Comments	TD	Files changed	Commit i - 1		
technical debt	be1d451225cc6d1e4bfe08ac7e67ad0780450f04	When adding a new feature, thought must be given to the long term technical debt +and maintenance that feature may require after inclusion. was removed and added to CONTRIBUTING.md	N				
	06d92d71a2ef414543940fd6ef1080176eb455bb	When adding a new feature, thought must be given to the long term technical debt and maintenance that feature may require after inclusion. was added to CONTRIBUTING.md[NF]	N				
debt	be1d451225cc6d1e4bfe08ac7e67ad0780450f04	Same as above	N				
	06d92d71a2ef414543940fd6ef1080176eb455bb	Same as above	N				
	f6aacbf6d3d73dd83601a396c45d130add60313						
	d8dda9de84eec34fa27dfb6be7c6bb0ab09de0d2						
may cause problem	a3874c7c3aeefc2330424f0247129a89c27edcd8	Alternatively, pass `--with-libressl` to `./configure`, however as the warning says, this is NOT supported, and may cause problems syncing the chain, or the node to fork off the network in unexpected circumstances. was removed from doc/build-openbsd.md	N				
	9639eadd93d988e9ea10e68ef38b21b33f76dc41	Alternatively, pass `--with-libressl` to `./configure`, however as the warning says, this is NOT supported, and may cause problems syncing the chain, or the node to fork off the network in unexpected circumstances. was added to doc/build-openbsd.md[NF]	N				
workaround for bug							
temporary solution							
this is wrong							
something bad is going on							
this indicates a more fundamental problem							
something serious is wrong							
doubt that this would work							
risk of this blowing up							
remove this code							

remove me	2e5361b9c20517a22a4d1fdae3077d23800e cba7	assert(0); //TODO remove me was removed from src/limitedmap.h		src/limitedmap.h	--		
	5ffc299404f4bcce3b9b2522096f8ed53d3181 d5	assert(0); //TODO remove me was added to src/limitedmap.h[NF]	Y	src/limitedmap.h			
	8269a0953ee9ccbd422433fc37184e60f94b 178	// Remove message queue was removed from src/qt/qtipcserver.cpp[DF]	N				
	41c938eede3e9ea29ef86f3c25bab8fa6a51a 509	// Remove message queue was added to src/qt/qtipcserver.cpp	N				
	65c82be0214c9480f12782284f9404dd7a6c1 041	Remove mentions on anonymity in package descriptions and manpage. was added to contrib/debian/changelog	N				
remove me before production							
it doesn't work yet							
doesn't work yet							
Fix Later	db6047d61b742be07442f891e70570b791c5 85e3	Of course, the subsidy is high enough, and transaction volume low + // enough, that fee sniping isn't a problem yet, but by implementing a fix + // now we ensure code won't be written that makes assumptions about + // nLockTime that preclude a fix later. was added to src/wallet/wallet.cpp	N				
revise later							
redo later							
reframe later							
redesign later							
save for later	b93974c3f32a5398a94db275cd23b7f7f8f269 bf	Again, either inv to all nodes or save for later was added to qa/rpc- tests/comptool.py[NF]	N	qa/rpc-tests/comptool.py			
Fix in the future							
revise in the future							
save for the future							
redo in the future							
reframe in the future							
redesign in the future							
is problematic	1581ecbc33edd9d6257e50b11d8854fbcca8f8 ad8	This file is problematic for out-of-tree builds if it exists. was removed from Makefile.am	N	Makefile.am			

	340012d9c978901d23ccedad5805c6630c49e756	This file is problematic for out-of-tree builds if it exists. was added to Makefile.am	N	Makefile.am		
	c25e98186d0f716451ef000e55646d25e014f573	Motivation: it is problematic to compute the CRC of a string that +// contains embedded CRCs. Therefore we recommend that CRCs stored +// somewhere (e.g., in files) should be masked before being stored. was added to util/crc32c.h[NF]	N	util/crc32c.h		
	64753f0438f66c51ba7ae2173120ae97f22d186e	Motivation: it is problematic to compute the CRC of a string that -// contains embedded CRCs. Therefore we recommend that CRCs stored -// somewhere (e.g., in files) should be masked before being stored. was removed from src/leveldb/util/crc32c.h[DF]	N	src/leveldb/util/crc32c.h		
	5e650d6d2dbfc284c300668e71188e663d8f0a45	Motivation: it is problematic to compute the CRC of a string that +// contains embedded CRCs. Therefore we recommend that CRCs stored +// somewhere (e.g., in files) should be masked before being stored. was added to src/leveldb/util/crc32c.h[NF]	N	src/leveldb/util/crc32c.h		
hack/hacky/hackery	5c66d41b7ffe8557ce5036613c9854d2f1dc3c37	// Hack to assume either it's completely dependent on other mempool txs or not at all was removed and added back to the file src/test/test_bitcoin.cpp	Y	src/test/test_bitcoin.cpp		
	e736772c56a883e2649cc8534dd7857a0718ec56	// Dirty hack to jump to BLOCKTXN code (TODO: move message handling into their own functions) and // Dirty hack to process as if it were just a headers message (TODO: move message handling into their own functions) was removed from src/main.cpp. 2) // Dirty hack to jump to BLOCKTXN code (TODO: move message handling into their own functions) and // Dirty hack to process as if it were just a headers message (TODO: move message handling into their own functions) was added to src/net_processing.cpp[NF]	Y	src/main.cpp src/net_processing.cpp	--	
	d25cd3ec4e8961c5f36c29a65395f52d0db294c5	// Dirty hack to jump to BLOCKTXN code (TODO: move message handling into their own functions) and // Dirty hack to process as if it were just a headers message (TODO: move message handling into their own functions) were added to src/main.cpp	Y	src/main.cpp		
	9d253620874087d96d64d0c4c771a582d9774657	#TODO: This is a quick hack that disables symbol checking for arm. + # Instead, we should investigate why these are popping up. was added to contrib/gitian-descriptors/gitian-linux.yml	Y	contrib/gitian-descriptors/gitian-linux.yml		
	5eeb913d6cff9cfe9a6769d7efe4a7b9f23de0f4	// Very ugly hack: as the global constructs and destructors run single + // threaded, we use this boolean to know whether LockData still exists, + // as DeleteLock can get called by global CCriticalSection destructors + // after LockData disappears. was added to src/sync.cpp	Y	src/sync.cpp		
	fb8a8cf2e610920e9eee61c19ed6080af064b43	A bit of a hack - dependency on a function defined in rpc/blockchain.cpp was added to	Y	src/rest.cpp		
	c0353064ddf71ad103bd19f6e7c10ff8e240ac46	// Hack to assume either its completely dependent on other mempool txs or not at all was added to src/test/test_bitcoin.cpp	Y	src/test/test_bitcoin.cpp		
	2e31d74b715515c344ba50f574831d6a73302aac	Ubuntu precise hack: Not an issue in later versions. was removed from contrib/gitian-descriptors/gitian-linux.yml		contrib/gitian-descriptors/gitian-linux.yml	--	
	6e182686163ce3c15b878bd78c41d8d18db344f1	/* Hack to initialize sig with a correctly-parsed but invalid signature. */ was added to src/pubkey.cpp	Y	src/pubkey.cpp		
	1d84107924ab36e38092cae43f6ad50dd5ead9ed	/* Hack to initialize sig with a correctly-parsed but invalid signature. */ was added to contrib/lax_der_parsing.c	Y	contrib/lax_der_parsing.c		
	b93974c3f32a5398a94db275cd23b7ff78f269bf	# Send ping and wait for response -- synchronization hack was added to qa/rpc-tests/comptool.py[NF]	Y	qa/rpc-tests/comptool.py		
	15facb4aca75122b6ae0dcc6f6e112127e6a0e59	Replace the temporary file hack currently used to change Bitcoin-QT's dock icon (OS X) with a buffer-based solution was added to doc/release-notes/release-notes-0.10.0.md	N	doc/release-notes/release-notes-0.10.0.md		
	c95ac83e5115a2e9f1c29df6a3839d6788b4eafd	# Ubuntu precise hack: Not an issue in later versions. was added to contrib/gitian-descriptors/gitian-linux.yml	Y	contrib/gitian-descriptors/gitian-linux.yml		
	1aead42d41178d0eeae2c1b7d0131af9d6fffd2	Hack in the use of our llvm headers rather than grabbing the old llvm-gcc. was removed from contrib/gitian-descriptors/gitian-osx-native.yml	Y	contrib/gitian-descriptors/gitian-osx-native.yml		

7014f382e3fd99226b7d844984e50706e2f67247	// ugly hack: for some unknown reason CAmount (instead of qint64) does not work here as expected was added to src/qt/bitcoinamountfield.h	Y	src/qt/bitcoinamountfield.h		
cbd22a50c4b3d2d0f133c5f8d34c9c7aca9c7f4f	The template matching params OP_SMALLDATA/etc are defined in opcode type enum + // as kind of implementation hack, they are "NOT" real opcodes. If found in real + // Script, just let the default: case deal with them. was added to src/script/script.cpp[NF] and removed from src/scriptutils.cpp	Y	src/script/script.cpp src/scriptutils.cpp	--	
da59f283350343a623820fa9ea48dd1ebb817064	# if this fails, go change the source as this is hacky and confusing! was added to contrib/devtools/update-translations.py 2) def postprocess_translations(reduce_diff_hacks=False);, if reduce_diff_hacks:, if reduce_diff_hacks: were also added	Y	contrib/devtools/update-translations.py		
bd0aa105198d3cc12751dc4645877af70cac6258	// write temp file hack (could also be done through QIODevice [memory]) was removed from src/qt/macdockiconhandler.mm		src/qt/macdockiconhandler.mm	--	
aab2c0fd7e7f629b20192ab9f05b281e5783cacc	The template matching params OP_SMALLDATA/etc are defined in opcode type enum + // as kind of implementation hack, they are "NOT" real opcodes. If found in real + // Script, just let the default: case deal with them. was added to src/script.cpp	Y	src/script.cpp		
1a97b22b9c8ca6f1c3c7e3285e1b98893691d421	# Hack in the use of our llvm headers rather than grabbing the old llvm-gcc. was added to contrib/gitian-descriptors/gitian-osx-native.yml	Y	contrib/gitian-descriptors/gitian-osx-native.yml		
7007402956579ace12d45cdcfae908802d3d6b6d	#define HTML_HACK_SP " " and #define THIN_SP_HTML HTML_HACK_SP were added to src/qt/bitcoinunits.h	Y	src/qt/bitcoinunits.h		
69baec96473ca2dc28ac016093770a46650a4ac9	This change is a short-term hack, to more quickly hunt for useful P2P connections. Better "leaf node" logic is in the works, but this should let us limp along until then. was added to doc/release-notes/release-notes-0.3.23.md	N	doc/release-notes/release-notes-0.3.23.md		
983cef48027f1bdf651c19d72cf615cf4345dbe6	// Todo: This is a hack, should be replaced with a cleaner solution! was removed from src/qt/walletmodel.h		src/qt/walletmodel.h	--	
c6c97e0f4e3d9d8d5b8a1e60174e9ff11e70b6c1	// If from an insecure payment request, this is used for storing + // the addresses, e.g. address-A address-B address-C. + // Info: As we don't need to process addresses in here when using + // payment requests, we can abuse it for displaying an address list. + // Todo: This is a hack, should be replaced with a cleaner solution! AND // If from a payment request, this is used for storing the memo + // Todo: This is a hack, should be replaced with a cleaner solution! were added to src/qt/walletmodel.h	Y	src/qt/walletmodel.h		
35b8af92265ed74de63c3818e5290c27b3f35df2	# Gross ugly hack that depends on qmake internals, unfortunately there is no other way to do it. was removed from bitcoin-qt.pro[DF]		bitcoin-qt.pro		
25c0cce7fb494fcb871d134e28b26504d30e34d3	// write temp file hack (could also be done through QIODevice [memory]) was added to src/qt/macdockiconhandler.mm	Y	src/qt/macdockiconhandler.mm		
c25e98186d0f716451ef000e55646d25e014f573	We provide our own simple hash table since it removes a whole bunch +// of porting hacks and is also faster than some of the built-in hash +// table implementations in some of the compiler/runtime combinations +// we have tested. was added to util/cache.cc	N	util/cache.cc		
64753f0438f66c51ba7ae2173120ae97f22d186e	We provide our own simple hash table since it removes a whole bunch -// of porting hacks and is also faster than some of the built-in hash -// table implementations in some of the compiler/runtime combinations -// we have tested. was removed from src/leveldb/util/cache.cc[DF]	N	src/leveldb/util/cache.cc		
b1024662eafddd5560fbfbac29333e5e967ca0f8	Gross ugly hack that depends on qmake internals, unfortunately there is no other way to do it. was removed and added to bitcoin-qt.pro		bitcoin-qt.pro		
4ca60bba5c3b8394c252715c71e8758d63c316ee	# Gross ugly hack that depends on qmake internals, unfortunately there's no other way to do it. was removed and added to bitcoin-qt.pro		bitcoin-qt.pro		
3ff3a2bd60a905d830de157390acde14aa0ffe3	# Gross ugly hack that depends on qmake internals, unfortunately there's no other way to do it. was added to bitcoin-qt.pro		bitcoin-qt.pro		
5e650d6d2dbfc284c300668e71188e663d8f0a45	We provide our own simple hash table since it removes a whole bunch +// of porting hacks and is also faster than some of the built-in hash +// table implementations in some of the compiler/runtime combinations +// we have tested. was added to src/leveldb/util/cache.cc[NF]	N	src/leveldb/util/cache.cc		

565c4771b6eba0eeb82f8602735100bbcf4b053e	// Hack to set wxConvLibc codepage to UTF-8 on Windows, - // may break if wxMBConv_win32 implementation in strconv.cpp changes. was removed from src/ui.cpp[DF]		src/ui.cpp	ee1d6e4ed0a931defc6f6273cc67a0a3e980940f	--	
b26141e2c5836ccd20bd32a9c653cf2ba5e5b5ee	/* hack nonce into lowest byte of w3 */ AND /*w3 = _mm_set1_epi32(ln[3]); nonce will be later hacked into the hash were removed from sha256.cpp [DF]	?	sha256.cpp			
0a61b0df1224a5470bcddab302bc199ca5a9e356	/*w3 = _mm_set1_epi32(ln[3]); nonce will be later hacked into the hash AND /* hack nonce into lowest byte of w3 */ were removed from sha256.cpp(?) and // Hack to set wxConvLibc codepage to UTF-8 on Windows, - // may break if wxMBConv_win32 implementation in strconv.cpp changes. was removed from and added to ui.cpp	Y	ui.cpp			
6ac7f9f144757f5f1a049c059351b978f83d1476	/*w3 = _mm_set1_epi32(ln[3]); nonce will be later hacked into the hash AND /* hack nonce into lowest byte of w3 */ were added to sha256.cpp [NF]	?	sha256.cpp			NOTE [1]
4bd188c4383d6e614e18f79dc337fbabe8464c82	/*w3 = _mm_set1_epi32(ln[3]); nonce will be later hacked into the hash AND /* hack nonce into lowest byte of w3 */ were added to sha256.cpp [NF]	?	sha256.cpp			
b6dc3b517b74ef2baa0e707f837a65b4ff5a6f2e	// Hack to set wxConvLibc codepage to UTF-8 on Windows, - // may break if wxMBConv_win32 implementation in strconv.cpp changes. was removed from init.cpp and added to ui.cpp	Y	ui.cpp cpp	init.	--	NOTE [2]
793200e5e918a6e993ddf701967a8b6535a6828f	// Hack to set wxConvLibc codepage to UTF-8 on Windows, - // may break if wxMBConv_win32 implementation in strconv.cpp changes. was removed from init.cpp and added to ui.cpp	Y	ui.cpp cpp	init.	--	
8bd66202c324a6c7a79abc0f1f0558dacbc59460	// Hack to set wxConvLibc codepage to UTF-8 on Windows, - // may break if wxMBConv_win32 implementation in strconv.cpp changes was removed from and added to init.cpp	Y	init.cpp			
c2430126d75318508836a24f90d3524320190465	// Hack to set wxConvLibc codepage to UTF-8 on Windows,*M + // may break if wxMBConv_win32 implementation in strconv.cpp changes. was added to init.cpp[NF] and removed from ui.cpp	Y	ui.cpp cpp	init.	--	NOTE [3]
5eede9d4d9932969cd75a47dc0d6fd4c090f0342	// Hack to set wxConvLibc codepage to UTF-8 on Windows,*M + // may break if wxMBConv_win32 implementation in strconv.cpp changes. was added to init.cpp[NF] and removed from ui.cpp	Y	ui.cpp cpp	init.	--	
e4806597650d8a17f1da7a5b5501eb367ea786fc	// Hack to change the (c) character from UTF-8 back to ANSI was removed from ui.cpp and // Hack to set wxConvLibc codepage to UTF-8 on Windows,*M + // may break if wxMBConv_win32 implementation in strconv.cpp changes. was added to ui.cpp	Y	ui.cpp			NOTE [4]
bcd27140384ec3703f3fd18ec89b0046a0ef17d3	// Hack to change the (c) character from UTF-8 back to ANSI was removed from ui.cpp and // Hack to set wxConvLibc codepage to UTF-8 on Windows,*M + // may break if wxMBConv_win32 implementation in strconv.cpp changes. was added to ui.cpp	Y	ui.cpp			
5253d1ab77fab1995ede03fb934edd67f1359ba8	// I'm tired of maintaining this hack that's only called by unfinished unused code. was removed from ui.cpp		ui.cpp		--	NOTE [5]
cc4b78d59f566ff43881f57797a16ce45eb1b80e	// I'm tired of maintaining this hack that's only called by unfinished unused code. was removed from ui.cpp		ui.cpp		--	
98500d70a8cf25af4bab80526fd128ccdc36ceeb	// Hack to change the (c) character from UTF-8 back to ANSI was added to ui.cpp	Y	ui.cpp			NOTE [6]
22f721dbf23cf5ce9e3ded9bcfb65a3894cc0f8c	// Hack to change the (c) character from UTF-8 back to ANSI was added to ui.cpp	Y	ui.cpp			
53d508072b02d522371bde148dcc3e925f472be7	// I'm tired of maintaining this hack that's only called by unfinished unused code, - // but I'm not willing to delete it because it serves as documentation of what the - // unfinished code was trying to do. was changed to // I'm tired of maintaining this hack that's only called by unfinished unused code. in ui.cpp	Y	ui.cpp			NOTE [7]

	2939cab06d65548abdaeb1ff218550eb47b93362	// I'm tired of maintaining this hack that's only called by unfinished unused code, - // but I'm not willing to delete it because it serves as documentation of what the - // unfinished code was trying to do. was changed to // I'm tired of maintaining this hack that's only called by unfinished unused code. in ui.cpp	Y	ui.cpp			
	8be979d9aef606f2a196a2addcf80473d571185b	// I'm tired of maintaining this hack that's only called by unfinished unused code,^M + // but I'm not willing to delete it because it serves as documentation of what the^M + // unfinished code was trying to do. was added to ui.cpp	Y	ui.cpp			NOTE [8]
	9e56521b4b79a90cfae63c658ed2ecb67acebfca	// I'm tired of maintaining this hack that's only called by unfinished unused code,^M + // but I'm not willing to delete it because it serves as documentation of what the^M + // unfinished code was trying to do. was added to ui.cpp	Y	ui.cpp			
	8dca7864f793072701f810e4c5ea12a6e1087085	NO MENTION OF HACK(?)					
	e00d4805d71de8bb32c9af600f5bd8fce2414d5d	NO MENTION OF HACK(?)					
	e071a3f6c06f41068ad17134189a4ac3073ef76b	NO MENTION OF HACK(?)					
	4405b78d6059e536c36974088a8ed4d9f0f29898	NO MENTION OF HACK(?)					
causes issue							
cause for issue							
this can be a mess							
this is temporary and will go away							
hope everything will work							
workaround	5c66d41b7ffe8557ce5036613c9854d2f1dc3c37	// Workaround for libunivalue pretty printer, which puts a space between comma's and newlines was changed to // Workaround for libunivalue pretty printer, which puts a space between commas and newlines in src/test/script_tests.cpp	Y	src/test/script_tests.cpp			
	89a3723bdc5d563f036c1a579c3d3539cc8fc6e7	// workaround for Qt OSX Bug: was removed from and added to src/qt/guiutil.h	Y	src/qt/guiutil.h			
	9358893518a19f38eab9186c768ad81c7a2f9cec	#7487 `00d57b4` Workaround Travis-side CI issues (luke-jr) was added to doc/release-notes/release-notes-0.12.1.md +- #7256 `08ab906` Add note to coin control dialog QT5 workaround (fanquake) +- #7334 `e1060c5` coincontrol workaround is still needed in qt5.4 (fixed in qt5.5) (MarcoFalke) +- #8139 `ae5575b` Fix interrupted HTTP RPC connection workaround for Python 3.5+ (sipa) were added to doc/release-notes/release-notes-0.13.0.md	N	doc/release-notes/release-notes-0.12.1.md doc/release-notes/release-notes-0.13.0.md			
	eb0c52ec1d3ffd4e2d78b2044c1c3d62de7711d3	#workaround for https://github.com/travis-ci/travis-ci/issues/5227 was removed from .travis.yml		.travis.yml			
	9a22a6c0891256f02f4906c1c13fb22a9722ec7c	ourselves (to workaround an issue with closing an asyncore socket when was removed and added to qa/rpc-tests/test_framework/mininode.py	Y	qa/rpc-tests/test_framework/mininode.py			

695041e4952ea40e02e3b0431d04ca6af7ec9711	// Workaround OSX linker warning: xcode uses different default symbol +// visibilities for static libs vs executables (see issue #25) was added to src/tinyformat.h	Y	src/tinyformat.h		
330b0f31ee5719d94f9e52dfc83c5d82168241f9	Workaround: + # Can either change the tip, or change the mempool and wait 5 seconds + # to trigger a recomputation of getblocktemplate. was added to qa/rpc-tests/p2p-segwit.py	Y	qa/rpc-tests/p2p-segwit.py		
605e8473a7ddca13b24a4020c7bd630aa5d374e2	scriptSigRet << OP_0; // workaround CHECKMULTISIG bug and CScript result; result << OP_0; // pop-one-too-many workaround were removed and ret.push_back(valtype()); // workaround CHECKMULTISIG bug and std::vector<valtype> result; result.push_back(valtype()); // pop-one-too-many workaround were added to src/script/sign.cpp	Y	src/script/sign.cpp		
291f8aa5daf80eed56d6cefa3d410652b412150a	# comptool workaround: to make sure b64 is delivered, manually erase b64a from blockstore + self.test.block_store.erase(b64a.sha256) was added to qa/rpc-tests/p2p-fullblocktest.py	Y	qa/rpc-tests/p2p-fullblocktest.py		
2ca8962a095ee29709316d163508f26ab1c340ae	#workaround for https://github.com/travis-ci/travis-ci/issues/5227 was added to .travis.yml		.travis.yml		
c7c664191fd6ca4843cfe9d00abf8f2362b9550f	// Workaround for libunivalue pretty printer, which puts a space between comma's and newlines was added to src/test/script_tests.cpp	Y	src/test/script_tests.cpp		
0e3ccbf26b11ea3d9ed7dfd39886d69097286e1	A temporary workaround +for previous versions of Bitcoin Core was to raise the default minimum +relay fee. was added to doc/release-notes/release-notes-0.12.0.md	N	doc/release-notes/release-notes-0.12.0.md		
9639eadd93d988e9ea10e68ef38b21b33f76dc41	There is no known workaround for this. was added to doc/build-openbsd.md	N	doc/build-openbsd.md		
ddf98d1d84343f2db0502a85628ef80f2ec57dbd	This is a workaround for https://bugs.python.org/issue3566 which is fixed in Python 3.5. was added to qa/rpc-tests/test_framework/authproxy.py	Y	qa/rpc-tests/test_framework/authproxy.py		
75a880390191aeaf7d2fa326f194349a891db022	#6286 `3902c15` Remove berkeley-db4 workaround in MacOSX build docs was added to doc/release-notes/release-notes-0.11.0.md	N	doc/release-notes/release-notes-0.11.0.md		
be31ff7d8ddb86d4499c7619ac2ec3a59434a596	The Ubuntu PPA from https://launchpad.net/~bitcoin/+archive/ubuntu/bitcoin +- A third-party or self-compiled Bitcoin Core + +upgrade to Bitcoin Core 0.9.4, which includes a workaround, **before** updating +OpenSSL. was added to doc/release-notes/release-notes-0.9.4.md	N	doc/release-notes/release-notes-0.9.4.md		
5d34e16d3a1e6ef37d5e6d254063c342b0e5fe39	self.nodes[i].getnewaddress("") # bug workaround, coins generated assigned to first getnewaddress! was added to qa/rpc-tests/txn_clone.py	Y	qa/rpc-tests/txn_clone.py		
ef3281750d9698ed58e07e72232d1604652f5fff	# Keep our own socket map for asyncore, so that we can track disconnects +# ourselves (to workaround an issue with closing an asyncore socket when +# using select) AND # We check for whether to disconnect outside of the asyncore + # loop to workaround the behavior of asyncore when using + # select were added to qa/rpc-tests/mininode.py	Y	qa/rpc-tests/mininode.py		
15facb4aca75122b6ae0dcc6f6e112127e6a0e59	`6093aa1` Qt: QProgressBar CPU-Issue workaround was added to doc/release-notes/release-notes-0.10.0.md[NF]	N	doc/release-notes/release-notes-0.10.0.md		
ffe3291466b5760e4179d604c54093717b127cc5	Workaround for Darwin assembler quirk. was removed from depends/patches/gmp/darwin_gmp_build_fix.patch[DF]		depends/patches/gmp/darwin_gmp_build_fix.patch		

1aead42d41178d0eeae2c1b7d0131af9d6fffd2	# Workaround: Upstream boost dev refuses to include patch that would allow Free Software cross-compile toolchain to work - # This patch was authored by the Fedora package developer and ships in Fedora's mingw32-boost. AND # Bug Workaround: boost-1.54.0 broke the ability to disable zlib, still broken in 1.55 were removed from contrib/gitian-descriptors/boost-win.yml[DF] AND # Workaround to prevent re-configuring by make; make all files have a date in the past (occurring 3 times) was removed from contrib/gitian-descriptors/deps-linux.yml[DF] AND # Workaround to prevent re-configuring by make (resulting in missing m4 error); make all files have a date in the past was removed from contrib/gitian-descriptors/deps-win.yml[DF]		contrib/gitian-descriptors/boost-win.yml contrib/gitian-descriptors/deps-linux.yml contrib/gitian-descriptors/deps-win.yml			
8656dbb095bc392d8d348bb68124696a03397f8	self.nodes[i].getnewaddress("") # bug workaround, coins generated assigned to first getnewaddress! was added to qa/rpc-tests/txn_doublepend.py[NF]	Y	qa/rpc-tests/txn_doublepend.py			
0ceab00d16a1d23da06a748889db25662b16bfb3	// workaround for Qt OSX Bug: was removed from src/qt/bitcoingui.cpp and added to src/qt/guiutil.h	Y	src/qt/bitcoingui.cpp src/qt/guiutil.h			
ff1e5ba8c75254ccfb08f408c607e98859d88a74	Workaround for Darwin assembler quirk. was added to depends/patches/gmp/darwin_gmp_build_fix.patch[NF]		depends/patches/gmp/darwin_gmp_build_fix.patch			
6093aa1bb03b03331cc2f48d8f6b749f8817c016	// workaround for Qt OSX Bug: was added to src/qt/bitcoingui.cpp	Y	src/qt/bitcoingui.cpp			
e088d65acbb181220b64a45a2a62bb79c0e40304	scriptSigRet << OP_0; // workaround CHECKMULTISIG bug AND CScript result; result << OP_0; // pop-one-too-many workaround were added to src/script/sign.cpp[NF] AND the same comments were removed from src/scriptutils.cpp	Y	src/script/sign.cpp src/scriptutils.cpp			
3f7a61fc0966bc63dfcafd231c449d922abb884c	-Then set up LXC and the rest with the following is a complex jumble of settings and workarounds: was changed to Then set up LXC and the rest with the following, which is a complex jumble of settings and workarounds: in doc/gitian-building.md	N	doc/gitian-building.md			
7007402956579ace12d45ccdfae908802d3d6b6d	// QMessageBox seems to have a bug whereby it doesn't display thin/hair spaces +// correctly. Workaround is to display a space in a small font. If you +// change this, please test that it doesn't cause the parent span to start +// wrapping. was added to src/qt/bitcoinunits.h	Y	src/qt/bitcoinunits.h			
2c47a00b5f5120f206b93574bae2bdc77538bf1d	Then set up LXC and the rest with the following is a complex jumble of settings and workarounds: was added to doc/gitian-building.md[NF]	N	doc/gitian-building.md			
26e4b52712faf8e60814f7f6030b6dded1df8038	Some ports also seem to obey either 'build_arch' or '-macosx_deployment_target', but not both at the same time. For example, building -on an OS X 10.6 64-bit machine fails. Official release builds of Bitcoin-Qt are -compiled on an OS X 10.6 32-bit machine to workaround that problem. was removed from doc/build-osx.md	N	doc/build-osx.md			
c43d1e5549f548efd221043bbc8069700ee22183	# Workaround to prevent re-configuring by make (resulting in missing m4 error); make all files have a date in the past was added to contrib/gitian-descriptors/deps-win.yml		contrib/gitian-descriptors/deps-win.yml			
aa9348563c88b984d9c42c41aa5d227145c8ac4b	# Workaround to prevent re-configuring by make; make all files have a date in the past was added 3 times to contrib/gitian-descriptors/deps-linux.yml		contrib/gitian-descriptors/deps-linux.yml			
65615a3a784c01128b408a915ab375c35640fb0a	# Qt: workaround for determinism in resource ordering was added to contrib/gitian-descriptors/gitian-win.yml and contrib/gitian-descriptors/qt-win.yml		contrib/gitian-descriptors/gitian-win.yml contrib/gitian-descriptors/qt-win.yml			
69baec96473ca2dc28ac016093770a46650a4ac9	Reannounce UPnP port forwards every 20 minutes, to workaround routers expiring old entries, and allow the -upnp option to override any stored setting. was added to doc/release-notes/release-notes-0.4.4.md[NF]. The same comment was added to doc/release-notes/release-notes-0.5.3.md[NF]. Run Bitcoin-Qt with the -server command-line option to workaround. was added to doc/release-notes/release-notes-0.8.2.md[NF]. Workaround for improbable memory access violation.was added to doc/release-notes/release-notes-0.5.4.md[NF]		doc/release-notes/release-notes-0.4.4.md doc/release-notes/release-notes-0.5.3.md doc/release-notes/release-notes-0.8.2.md doc/release-notes/release-notes-0.5.4.md			

	b77dfdc9e36e308aa806d63aa3b5628971789d5a	# define TINYFORMAT_OLD_LIBSTDCPLUSPLUS_WORKAROUND + #ifdef TINYFORMAT_OLD_LIBSTDCPLUSPLUS_WORKAROUND + struct formatZeroIntegerWorkaround + struct formatZeroIntegerWorkaround<T,true> + #endif // TINYFORMAT_OLD_LIBSTDCPLUSPLUS_WORKAROUND + #ifdef TINYFORMAT_OLD_LIBSTDCPLUSPLUS_WORKAROUND + else if(detail::formatZeroIntegerWorkaround<T>::invoke(out, value)) + /*; were added to the src/tinyformat.h[NF]	Y	src/tinyformat.h		
	f622232bcf8ebc7398f4c8988711155c22705fd4	# Workaround: Upstream boost dev refuses to include patch that would allow Free Software cross-compile toolchain to work AND # Bug Workaround: boost-1.54.0 broke the ability to disable zlib, still broken in 1.55 were added to contrib/gitian-descriptors/boost-win.yml[NF] and removed from contrib/gitian-descriptors/boost-win32.yml[DF]		contrib/gitian-descriptors/boost-win.yml contrib/gitian-descriptors/boost-win32.yml		
	56fce1fdbb688e9c48ba689d8f69bf66501d91c5	# Bug Workaround: boost-1.54.0 broke the ability to disable zlib was changed to # Bug Workaround: boost-1.54.0 broke the ability to disable zlib, still broken in 1.55 in contrib/gitian-descriptors/boost-win32.yml		contrib/gitian-descriptors/boost-win32.yml		
	f2bc1d35d6900e044209e76533dc514d31f768fe	# Workaround: Upstream boost dev refuses to include patch that would allow Free Software cross-compile toolchain to work AND # Bug Workaround: boost-1.54.0 broke the ability to disable zlib were added to contrib/gitian-descriptors/boost-win32.yml		contrib/gitian-descriptors/boost-win32.yml		
	35b8af92265ed74de63c3818e5290c27b3f35df2	# This is a workaround for Ubuntu bug #691722, the default -fstack-protector causes - # -fstack-protector-all to be ignored unless -fno-stack-protector is used first. was removed from src/makefile.unix[DF]		src/makefile.unix		
	68cb1cd6b6945c9068b0cde43fb8227819b3c45	Official release builds of Bitcoin-Qt are + compiled on an OS X 10.6 32-bit machine to workaround that problem. was added to doc/build-osx.md[NF]	N	doc/build-osx.md		
	382b75f528fd3d2d3ac76076e8b3325d9d471f96	// Override style sheet for progress bar for styles that have a segmented progress bar, + // as they make the text unreadable (workaround for issue #1071) was added to /src/qt/bitcoingui.cpp	Y	/src/qt/bitcoingui.cpp		
	fc4743faa8cefb0eaeceec51f6c8984a920c96d73	["It caught a bug in the workaround for 23b397edccd3740a74adb603c9756370fafcde9bcc4483eb271ecad09a94dd63 in an overly simple implementation"] was added to src/test/data/tx_valid.json[NF]	N	src/test/data/tx_valid.json		
	a2709fad7f57b000333371954016045e12fc4bed	CScript result; result << OP_0; // pop-one-too-many workaround was added to src/script.cpp	Y	src/script.cpp		
	460b66b14b80a55279d837c91aff1f1c3ca60d7f	if (AffinityBugWorkaround(ThreadBitcoinMiner)) - return; was removed from src/main.cpp. -inline bool AffinityBugWorkaround(void(*pfn)(void*)) AND printf ("AffinityBugWorkaround() : SetThreadAffinityMask=%d, ProcessAffinityMask=%d, restarting thread\n", dwPrev2, dwProcessAffinityMask); were removed from src/util.h		src/main.cpp src/util.h	--	
	7f10522c4316abb9d7dc6c97e48436a22b1a19a1	// Workaround for "multiple definition of `_tls_used'" was removed from src/util.cpp		src/util.cpp	--	
	e679ec969c8b22c676ebb10bea1038f6c8f13b33	// CHECKMULTISIG bug workaround: was removed AND scriptSigRet << OP_0; // workaround CHECKMULTISIG bug was added to src/script.cpp	Y	src/script.cpp		
	bf798734db4539a39edd6badf54a1c3aecf193e5	// CHECKMULTISIG bug workaround: was added to src/script.cpp AND result << OP_0; // CHECKMULTISIG bug workaround was added to src/test/multisig_tests.cpp[NF]	Y	src/script.cpp src/test/multisig_tests.cpp		
	f6aacbf6d3d73dd83601a396c45d130add60313	# Workaround a bug in macdeployqt: https://bugreports.qt.nokia.com/browse/QTBUG-21913 -# (when fixed, this won't be necessary) was removed from contrib/create_osx_dmg.sh[DF]		contrib/create_osx_dmg.sh	25e65733178606eb95e577a634d576f54ead35c4	--
	1e5f9393818cdc2264e7e44ed17291366da3dcb9	# This is a workaround for Ubuntu bug #691722, the default -fstack-protector causes + # -fstack-protector-all to be ignored unless -fno-stack-protector is used first. was added to src/makefile.unix		src/makefile.unix		
	b50ac8f73e3e3aa47be567f3fcd5b7c8933c49db	# Workaround a bug in macdeployqt: https://bugreports.qt.nokia.com/browse/QTBUG-21913 +# (when fixed, this won't be necessary) was added to contrib/create_osx_dmg.sh	Y	contrib/create_osx_dmg.sh		

6ccff2cbdebca38e4913b679784a4865edfbb12a	fixed incomplete workaround of min/max conflict in MSVC AND added compatibility fixes/workarounds for STLport 4.5, GCC 3.0.2, were removed from src/cryptopp/Readme.txt. // Define this to workaround a Microsoft CryptoAPI bug where -// each call to CryptAcquireContext causes a 100 KB memory leak. was removed from src/cryptopp/config.h[DF] - and MSVC 7.0. // VC60 workaround: it doesn't allow typename in some places AND // VC60 workaround: can't cast unsigned __int64 to float or double were removed from src/cryptopp/config.h[DF]. // VC60 workaround: using enums as template parameters causes problems was removed from src/cryptopp/cryptlib.h[DF]. // Sun Studio 11/CC 5.8 workaround: it generates incorrect code when casting to an empty virtual base class was removed from src/cryptopp/cryptlib.h[DF]. // VC60 workaround: no co-variant return type was removed from src/cryptopp/cryptlib.h[DF]. // VC2005 workaround: disable declarations that conflict with winnt.h was removed from src/cryptopp/misc.h[DF]. assert(a==0 a>0); // GCC workaround: get rid of the warning "comparison is always true due to limited range of data type" was removed from src/cryptopp/misc.h[DF]. inline unsigned int GetAlignmentOf(T *dummy=NULL) // VC60 workaround AND inline bool IsAligned(const void *p, T *dummy=NULL) // VC60 workaround were also removed from src/cryptopp/misc.h[DF]. #pragma warning(disable: 4700) // VC60 workaround: don't know how to get rid of this warning was removed from src/cryptopp/secblock.h[DF]. // VC60 workaround: prevent VC 6 from inlining this function was removed from src/cryptopp/sha.cpp[DF]		src/cryptopp/config.h src/cryptopp/cryptlib.h src/cryptopp/misc.h src/cryptopp/secblock.h src/cryptopp/sha.cpp	52e9fe929615d339732c060573b0637c51e353dc		
565c4771b6eba0eeb82f8602735100bbcf4b053e	strSort = " " + strSort; // leading space to workaround wx2.9.0 ubuntu 9.10 bug was removed from src/ui.cpp[DF]		src/ui.cpp	ee1d6e4ed0a931defc6f6273cc67a0a3e980940f	--	
829e21733b1c232c79c1a2d10598eb47b2f5f7f9	// NOTE: CHECKMULTISIG has an unfortunate bug; it requires + // one extra item on the stack, before the signatures. + // Putting OP_0 on the stack is the workaround; + // fixing the bug would mean splitting the blockchain (old + // clients would not accept new CHECKMULTISIG transactions, + // and vice-versa) was added to src/test/script_tests.cpp	Y	src/test/script_tests.cpp			
69d605f410c83235aa7b757445e7d0166fcfe2d9	if (AffinityBugWorkaround(ThreadBitcoinMiner)) was added to core/src/main.cpp[NF]	Y	core/src/main.cpp			
2097c09a9b5dda5bfa0de1906a8c155c3c3b529b	// Workaround for "multiple definition of `tls_used`" +// http://svn.boost.org/trac/boost/ticket/4258 was added to core/src/util.cpp[NF]	Y	core/src/util.cpp			
6644d98d9eeaa24c197d18665b1461b8cd4300ea	// Define this to workaround a Microsoft CryptoAPI bug where +// each call to CryptAcquireContext causes a 100 KB memory leak. AND // VC60 workaround: it doesn't allow typename in some places AND // VC60 workaround: can't cast unsigned __int64 to float or double AND were added to cryptopp/include/cryptopp/config.h[NF]. // VC60 workaround: using enums as template parameters causes problems AND // Sun Studio 11/CC 5.8 workaround: it generates incorrect code when casting to an empty virtual base class AND // VC60 workaround: no co-variant return type were added to cryptopp/include/cryptopp/cryptlib.h[NF]. // VC2005 workaround: disable declarations that conflict with winnt.h AND assert(a==0 a>0); // GCC workaround: get rid of the warning "comparison is always true due to limited range of data type" AND inline unsigned int GetAlignmentOf(T *dummy=NULL) // VC60 workaround AND inline bool IsAligned(const void *p, T *dummy=NULL) // VC60 workaround were added to cryptopp/include/cryptopp/misc.h[NF]. #pragma warning(disable: 4700) // VC60 workaround: don't know how to get rid of this warning was added to cryptopp/include/cryptopp/secblock.h[NF]. // VC60 workaround: prevent VC 6 from inlining this function was added to cryptopp/src/sha.cpp[NF]	Y	cryptopp/include/cryptopp/ config.h cryptopp/include/cryptopp/ cryptlib.h cryptopp/include/cryptopp/ misc.h cryptopp/include/cryptopp/ secblock.h cryptopp/src/sha.cpp			
1f2e0df86573910257318cbdc2c80d114f90d85c	inline bool AffinityBugWorkaround(void(*pfn)(void*)) AND printf ("AffinityBugWorkaround() : SetThreadAffinityMask=%d, ProcessAffinityMask=%d, restarting thread\n", dwPrev2, dwProcessAffinityMask) were added to lib/include/util.h[NF]	Y	lib/include/util.h			
5cbf75324d1509a1262b65c5073314a4da3f6d77	if (AffinityBugWorkaround(ThreadBitcoinMiner)) was added to main.cpp. inline bool AffinityBugWorkaround(void(*pfn)(void*)) AND printf ("AffinityBugWorkaround() : SetThreadAffinityMask=%d, ProcessAffinityMask=%d, restarting thread\n", dwPrev2, dwProcessAffinityMask); were added to util.h.	Y	main.cpp util.h			

	0a61b0df1224a5470bcddab302bc199ca5a9e356	fixed incomplete workaround of min/max conflict in MSVC AND added compatibility fixes/workarounds for STLport 4.5, GCC 3.0.2, - and MSVC 7.0 were removed from and added to cryptopp/Readme.txt 2) Define this to workaround a Microsoft CryptoAPI bug where -// each call to CryptAcquireContext causes a 100 KB memory leak AND // VC60 workaround: it doesn't allow typename in some places AND // VC60 workaround: can't cast unsigned __int64 to float or double were removed from and added back to cryptopp/config.h.3) // VC60 workaround: using enums as template parameters causes problems AND // Sun Studio 11/CC 5.8 workaround: it generates incorrect code when casting to an empty virtual base class - char m_sunCCworkaround; AND // VC60 workaround: no co-variant return type were removed and added back to cryptopp/cryptlib.h.4) // VC2005 workaround: disable declarations that conflict with winnt.h AND // GCC workaround: get rid of the warning "comparison is always true due to limited range of data type" AND // VC60 workaround were removed from and added back to cryptopp/misc.h 4) // VC60 workaround: don't know how to get rid of this warning was removed from and added to cryptopp/secblock.h 5) // VC60 workaround: prevent VC 6 from inlining this function was removed from and added to cryptopp/sha.cpp 6) // leading space to workaround wx2.9.0 ubuntu 9.10 bug was removed from and added to ui.cpp 7) #define WORKAROUND_MS_BUG_Q258000 was removed from and added to cryptopp/config.h	Y	cryptopp/Readme.txt cryptopp/config.h cryptopp/cryptlib.h cryptopp/misc.h cryptopp/secblock.h cryptopp/sha.cpp ui.cpp		
	6ac7f9f144757f5f1a049c059351b978f83d1476	// Workaround for "multiple definition of `tls_used'" +// http://svn.boost.org/trac/boost/ticket/4258 was added to util.cpp	Y	util.cpp		NOTE [9]
	4bd188c4383d6e614e18f79dc337fbabe8464c82	// Workaround for "multiple definition of `tls_used'"AM +// http://svn.boost.org/trac/boost/ticket/4258 was added to util.cpp	Y	util.cpp		
	3dd20ff2f87958a440c9dceae8abe3967aa48fcf	fixed incomplete workaround of min/max conflict in MSVC AND added compatibility fixes/workarounds for STLport 4.5, GCC 3.0.2, + and MSVC 7.0 were added to cryptopp/Readme.txt[NF] 2) // Define this to workaround a Microsoft CryptoAPI bug where +// each call to CryptAcquireContext causes a 100 KB memory leak. AND // VC60 workaround: it doesn't allow typename in some places AND // VC60 workaround: can't cast unsigned __int64 to float or double were added to cryptopp/config.h[NF] 3) // VC60 workaround: using enums as template parameters causes problems AND // Sun Studio 11/CC 5.8 workaround: it generates incorrect code when casting to an empty virtual base class AND char m_sunCCworkaround; AND // VC60 workaround: no co-variant return type were added to cryptopp/cryptlib.h[NF] 4) // VC2005 workaround: disable declarations that conflict with winnt.h AND // GCC workaround: get rid of the warning "comparison is always true due to limited range of data type" AND // VC60 workaround were added to cryptopp/misc.h[NF] 5) // VC60 workaround: don't know how to get rid of this warning AND // VC60 workaround: prevent VC 6 from inlining this function were added to cryptopp/sha.cpp[NF] 7) #define WORKAROUND_MS_BUG_Q258000 was added to cryptopp/config.h[NF]	Y	cryptopp/Readme.txt cryptopp/config.h cryptopp/cryptlib.h cryptopp/misc.h cryptopp/sha.cpp		NOTE [10]
	813505cc1313b7e191b787f93e573acb91fa1464	fixed incomplete workaround of min/max conflict in MSVC AND added compatibility fixes/workarounds for STLport 4.5, GCC 3.0.2, + and MSVC 7.0 were added to cryptopp/Readme.txt[NF] 2) // Define this to workaround a Microsoft CryptoAPI bug where +// each call to CryptAcquireContext causes a 100 KB memory leak. AND // VC60 workaround: it doesn't allow typename in some places AND // VC60 workaround: can't cast unsigned __int64 to float or double were added to cryptopp/config.h[NF] 3) // VC60 workaround: using enums as template parameters causes problems AND // Sun Studio 11/CC 5.8 workaround: it generates incorrect code when casting to an empty virtual base class AND char m_sunCCworkaround; AND // VC60 workaround: no co-variant return type were added to cryptopp/cryptlib.h[NF] 4) // VC2005 workaround: disable declarations that conflict with winnt.h AND // GCC workaround: get rid of the warning "comparison is always true due to limited range of data type" AND // VC60 workaround were added to cryptopp/misc.h[NF] 5) // VC60 workaround: don't know how to get rid of this warning AND // VC60 workaround: prevent VC 6 from inlining this function were added to cryptopp/sha.cpp[NF] 7) #define WORKAROUND_MS_BUG_Q258000 was added to cryptopp/config.h[NF]	Y	cryptopp/Readme.txt cryptopp/config.h cryptopp/cryptlib.h cryptopp/misc.h cryptopp/sha.cpp		

	8bd66202c324a6c7a79abc0f1f0558dacbc59460	strSort = " " + strSort; // leading space to workaround wx2.9.0 ubuntu 9.10 bug was removed from ui.cpp and added to the same file	Y	ui.cpp			
	e4806597650d8a17f1da7a5b5501eb367ea786fc	// Workaround until upgrade to wxWidgets supporting UTF-8 was removed from ui.cpp		ui.cpp		--	NOTE [11]
	bcd27140384ec3703f3fd18ec89b0046a0ef17d3	// Workaround until upgrade to wxWidgets supporting UTF-8 was removed from ui.cpp		ui.cpp		--	
	5253d1ab77fab1995ede03fb934edd67f1359ba8	vReviews.size(); // msvc workaround, just need to do anything with vReviews was rremoved from db.cpp		db.cpp		--	NOTE [12]
	cc4b78d59f566ff43881f57797a16ce45eb1b80e	vReviews.size(); // msvc workaround, just need to do anything with vReviews was rremoved from db.cpp		db.cpp		--	
	c4319e678f693d5fbc49bd357ded1c8f951476e9	strSort = " " + strSort; // leading space to workaround wx2.9.0 ubuntu 9.10 bug was added to ui.cpp	Y	ui.cpp			NOTE [13]
	7271c7ff35b74b7fa0bb9dd2db95b99fb9f9db01	strSort = " " + strSort; // leading space to workaround wx2.9.0 ubuntu 9.10 bug was added to ui.cpp	Y	ui.cpp			
	e071a3fc06f41068ad17134189a4ac3073ef76b	vReviews.size(); // msvc workaround, just need to do anything with vReviews was added to db.cpp[NF]. // Workaround until upgrade to wxWidgets supporting UTF-8 was added to ui.cpp[NF]	Y	db.cpp ui.cpp			NOTE [14]
	4405b78d6059e536c36974088a8ed4d9f0f29898	vReviews.size(); // msvc workaround, just need to do anything with vReviews was added to db.cpp[NF]. // Workaround until upgrade to wxWidgets supporting UTF-8 was added to ui.cpp[NF]	Y	db.cpp ui.cpp			
can cause problems							
this is temporary							
problem will go away							
in the future	c28583d0629bc6e12ea5d8c660c9b3586be5ad71	# Generate a block containing the initial transactions, then another + # block further in the future (past the rescan window). was added to qa/rpc-tests/import-rescan.py	N	qa/rpc-tests/import-rescan.py			
	d4781ac6c26399d316a2b25f656d9a4c0990c419	// We seem to be rather well-synced, so it appears pfrom was the first to provide us - // with this block! Let's get them to announce using compact blocks in the future. was removed from src/net_processing.cpp	N	src/net_processing.cpp			
	e736772c56a883e2649cc8534dd7857a0718ec56	// We seem to be rather well-synced, so it appears pfrom was the first to provide us with this block! Let's get them to announce using compact blocks in the future. was removed from src/main.cpp and added to src/net_processing.cpp	N	src/main.cpp src/net_processing.cpp			
	b26a7b5377f400760f49ba3aed468469ef27786e	/** In the future use NetworkIDString() for RPC fields */ was removed from src/chainparams.h	?	src/chainparams.h			
	02a337defdd854efc78ecba6d1fb19cb1c075f16	// We seem to be rather well-synced, so it appears pfrom was the first to provide us with this block! Let's get them to announce using compact blocks in the future. was added to src/main.cpp	N	src/main.cpp			
	248f3a76a825a332e5495c5947ad283a9e5e938f	Currently the only use of isInvalid is indicate uncompressed keys in SIGVERSION_WITNESS_V0 script, but could also be used in similar cases in the future was added to src/script/ismine.h	N	src/script/ismine.h			

f2e939bdbceef1ee90d59fab9ab99b944b4bce1	It is recommended to use HTML_EXTRA_STYLESHEET instead of this tag, as + # it is more robust and this tag (HTML_STYLESHEET) will in the future become + # obsolete. AND Compiled HTML + # files are now used as the Windows 98 help format, and will replace the old + # Windows help format (.hlp) on all Windows platforms in the future. were added to doc/Doxyfile	N	doc/Doxyfile		
eeeebdd3cba1e69835f826236f7d71ee8d76ace4	The string parameter is just an arbitrary name given to the default wallet. It's been prepended with a tilde to avoid name collisions in the future with additional wallets. was removed from doc/multiwallet-qt.md[DF]. /** Display name for default wallet name. Uses tilde to avoid name + * collisions in the future with additional wallets */ was added to src/qt/bitcoingui.cpp	N	doc/multiwallet-qt.md src/qt/bitcoingui.cpp		
9358893518a19f38eab9186c768ad81c7a2f9cec	Miners get to choose what time they use for their header time, with the +consensus rule being that no node will accept a block whose time is more +than two hours in the future. was added to doc/release-notes/release-notes-0.12.1.md[NF]. In the future, the '-blockmaxsize' option may be removed, as block creation is no longer optimized for this metric. was added to doc/release-notes/release-notes-0.13.0.md[NF]	N	doc/release-notes/release-notes-0.12.1.md doc/release-notes/release-notes-0.13.0.md		
122786d0e0170c73536360b705af711e1338adbf	consensus.BIP34Height = 100000000; // BIP34 has not activated on regtest (far in the future so block v1 are not rejected in tests) was added to src/chainparams.cpp	N	src/chainparams.cpp		
be1d451225cc6d1e4bfe08ac7e67ad0780450f04	Commit messages should be helpful to people reading your code in the future, so explain the reasoning for your decisions. was removed AND added back. If features get orphaned with no maintainer in the future, they may be removed by the Repository Maintainer. was also removed and added back to CONTRIBUTING.md	N	CONTRIBUTING.md		
f8528134fc188abc5c7175a19680206964a8fade	consensus.vDeployments[Consensus::DEPLOYMENT_SEGWIT].nStartTime = 2000000000; // Far in the future was removed from src/chainparams.cpp	N	src/chainparams.cpp		
8b49040854be2e26b66366aee1cba4716f93d93	consensus.vDeployments[Consensus::DEPLOYMENT_SEGWIT].nStartTime = 2000000000; // Far in the future was added to src/chainparams.cpp	N	src/chainparams.cpp		
2f34a2e476ae9d0585c67e275d238e44119c56cf	// We seem to be rather well-synced, so it appears pfrom was the first to provide us + // with this block! Let's get them to announce using compact blocks in the future. was added to src/main.cpp	N	src/main.cpp		
cbda71cf04ef6f2abe6eaa56c3140a6f5cff4feb	return state.Invalid(false, REJECT_INVALID, "time-too-new", "block timestamp too far in the future"); was removed and added to src/main.cpp	N	src/main.cpp		
291f8aa5daf80eed56d6cefa3d410652b412150a	# A block with timestamp > 2 hrs in the future was added to qa/rpc-tests/p2p-fullblocktest.py	N	qa/rpc-tests/p2p-fullblocktest.py		
87b8175d999e10082049afd1fb6440cf54d86f1e	to ELF recently, or will in the future. was changed to # to ELF recently (or will in the future) and ABI. in depends/config.guess	N	depends/config.guess		
19d73d540c8de4a73b5b2a05bebd762e74890a20	self.nodes[0].setmocktime(0) # set time back to present so yielded blocks aren't in the future as we advance last_block_time was added to qa/rpc-tests/bip68-112-113-p2p.py[NF]	N	qa/rpc-tests/bip68-112-113-p2p.py		
146746bbafe5dfea7b2019231c2d37fe57a4c3bb	Even if +that is the version your Linux distribution ships with, it probably is a good +idea to build Bitcoin Core against a static version of that library compiled +according to the instructions in the 'build-unix.md' file so that any changes +the distribution may make in the future will not result in a problem for users. was added to contrib/rpm/README.md[NF]	N	contrib/rpm/README.md		

0e3ccbfb26b11ea3d9ed7dfd39886d69097286e1	This release includes several changes related to the [BIP65]] soft fork +which redefines the existing OP_NOP2 opcode as OP_CHECKLOCKTIMEVERIFY +(CLTV) so that a transaction output can be made unspendable until a +specified point in the future. AND Miners get to choose what time they use for their header time, with the +consensus rule being that no node will accept a block whose time is more +than two hours in the future. were added to doc/release-notes/release-notes-0.11.2.md[NF]	N	doc/release-notes/release-notes-0.11.2.md		
93fc58c7426b5f3c68f2657626698846fb512ee8	return state.Invalid(error("CheckBlockHeader(): block timestamp too far in the future"), - REJECT_INVALID, "time-too-new"); was changed to return state.Invalid(false, REJECT_INVALID, "time-too-new", "block timestamp too far in the future"); in src/main.cpp	N	src/main.cpp		
5400ef6bcb9d243b2b21697775aa6491115420f3	/** Return a timestamp in the future (in microseconds) for exponentially distributed events. */ was added to src/net.h	N	src/net.h		
c12ff995f7d70aafb12f34887fb64aa7482bbc85	-Previously it was impossible to create a transaction output that was guaranteed -to be unspendable until a specific date in the future. was removed from doc/release-notes.md	N	doc/release-notes.md		
a264c32e3321ae909ca59cb8ce8bf5d812dbc4e1	In libevent + // master that appears to be solved, so in the future that solution + // could be used again (if desirable). was added to src/httpserver.cpp	N	src/httpserver.cpp		
c939792baaaa5f34ee4079b871d56054e36fb26b	Previously it was impossible to create a transaction output that was guaranteed +to be unspendable until a specific date in the future. was added to doc/release-notes.md	N	doc/release-notes.md		
06d92d71a2ef414543940fd6ef1080176eb455bb	Commit messages should be helpful to people reading your code in the future, so explain the reasoning for your decisions. was removed AND added back. If features get orphaned with no maintainer in the future, they may be removed by the Repository Maintainer. was also removed and added back to CONTRIBUTING.md	N	CONTRIBUTING.md		
75a880390191aeaf7d2fa326f194349a891db022	For now, block pruning disables block relay. In the future, nodes with block +pruning will at a minimum relay "new" blocks, meaning blocks that extend their +active chain. was added to doc/release-notes/release-notes-0.11.0.md [NF]	N	doc/release-notes/release-notes-0.11.0.md		
3e8ccc765a6d02b701ebc40e556e32e4c0613987	For now, block pruning disables block relay. In the future, nodes with block +pruning will at a minimum relay "new" blocks, meaning blocks that extend their +active chain. was removed from doc/release-notes.md	N	doc/release-notes.md		
44e1aad50089bc4c4fb72e476ff2e100c1241521	For now, block pruning disables block relay. In the future, nodes with block +pruning will at a minimum relay "new" blocks, meaning blocks that extend their +active chain. was added to doc/release-notes.md	N	doc/release-notes.md		
68d370bec44dd6c0b9bdc76308de1a96fe34e5b8	// Each task adds or subtracts from one of the counters a + // random amount, and then schedules another task 0-1000 + // microseconds in the future to subtract or add from + // the counter -random_amount+1, so in the end the shared + // counters should sum to the number of initial tasks performed. was added to src/test/scheduler_tests.cpp[NF]	N	src/test/scheduler_tests.cpp		
f9ec3f0fad11ee9889af977e16915f5d6e01944	// If we're missing data, then add back to mapBlocksUnlinked, + // so that if the block arrives in the future we can try adding + // to setBlockIndexCandidates again. was added to src/main.cpp	N	src/main.cpp		
06087bda87214e3f4995a0de65633511d4158d78	// other OSes: helpful when dealing with payment request files (in the future) was removed from src/qt/paymentserver.cpp	N	src/qt/paymentserver.cpp		

5262fde0ecd19f1febbfcd488f2be41c5dff047	return state.Invalid(error("CheckBlockHeader() : block timestamp too far in the future"), was changed to return state.Invalid(error("CheckBlockHeader(): block timestamp too far in the future"), in src/main.cpp	N	src/main.cpp		
409e9ef04cae479cda547500dfb1937633275b49	-Beware. This part is ugly. Very very very ugly. In the future, this should be -broken out into a new repository and cleaned up. was removed from doc/README_osx.txt	N	doc/README_osx.txt		
b9a36b15bff6a6fc0bb7f8bf6e7361dac3919e78	Mandatory script verification flags that all new blocks must comply with for -// them to be valid. (but old blocks may not comply with) Currently just P2SH, -// but in the future other flags may be added, such as a soft-fork to enforce -// strict DER encoding. was removed and added back to src/script/standard.h	N	src/script/standard.h		
f2e03ffae92ec1333c0eb724835ad56138f54716	/* In the future use NetworkIDString() for RPC fields */ was changed to /** In the future use NetworkIDString() for RPC fields */ in src/chainparams.h	?	src/chainparams.h		
341735eb8f42e898cf9d4d130709471e5d01abe2	Larger windows tolerate larger download speed differences between peer, but increase the potential + * degree of disordering of blocks on disk (which make reindexing and in the future perhaps pruning + * harder). We'll probably want to make this a per-peer adaptive value at some point. */ was added to src/main.h	N	src/main.h		
cc972107997122cb31f015b4e70f81781dbce784	/* In the future use NetworkIDString() for RPC fields */ was added to src/chainparams.h	?	src/chainparams.h		
c4408a6c8534a36711557028a69314024cbcb1bbe	// Mandatory script verification flags that all new blocks must comply with for +// them to be valid. (but old blocks may not comply with) Currently just P2SH, +// but in the future other flags may be added, such as a soft-fork to enforce +// strict DER encoding. was added to src/script/standard.h[NF] and removed from src/scriptutils.h	N	src/script/standard.h src/scriptutils.h		
1dec09b341f61836147d87656aea7f7be02aab6d	# The Operating System including object format, if it has switched + # to ELF recently, or will in the future. was added to depends/config.guess[NF]	N	depends/config.guess		
1a97b22b9c8ca6f1c3c7e3285e1b98893691d421	In order to build a working toolchain, the following source packages are needed +from Apple: cctools, dyld, and ld64. + +Beware. This part is ugly. Very very very ugly. In the future, this should be +broken out into a new repository and cleaned up. was added to doc/README_osx.txt[NF]	N	doc/README_osx.txt		
68f7d1d7af39a8ea6510f888e8e058e8e8faa007	// Mandatory script verification flags that all new blocks must comply with for +// them to be valid. (but old blocks may not comply with) Currently just P2SH, +// but in the future other flags may be added, such as a soft-fork to enforce +// strict DER encoding. was added to src/script.h	N	src/script.h		
942b33a19d3aa96326acc044c1834c48beff77c	return state.Invalid(error("CheckBlock() : block timestamp too far in the future"), was changed to return state.Invalid(error("CheckBlockHeader() : block timestamp too far in the future"), in src/main.cpp	N	src/main.cpp		
69baec96473ca2dc28ac016093770a46650a4ac9	However, this release includes a bug +fix that makes it a little bit more difficult for attackers to double-spend a +certain type ("lockTime in the future") of zero-confirmation transaction. was added to doc/release-notes/release-notes-0.8.0.md[NF]	N	doc/release-notes/release-notes-0.8.0.md		
f622232bcf8ebc7398f4c8988711155c22705fd4	# Note: Might need these options in the future for 64bit builds. was removed from contrib/gitian-descriptors/boost-win32.yml[DF]	N	contrib/gitian-descriptors/boost-win32.yml		

358ce2664d27da5bd8ae4c80655368219dcf18b3	return state.Invalid(error("CheckBlock() : block timestamp too far in the future")); was changed to return state.Invalid(error("CheckBlock() : block timestamp too far in the future"), + REJECT_INVALID, "time in future"); in src/main.cpp	N	src/main.cpp		
08dd1b7be172f7ea580dc9bdf20ac15ea9a2ed31	// other OSes: helpful when dealing with payment-request files (in the future) was removed and added back to src/qt/paymentserver.cpp	N	src/qt/paymentserver.cpp		
95d4a2be95a4b902c4cda4a8b57c7be9e572aee6	// Install global event filter to catch QFileOpenEvents + // on Mac: sent when you click bitcoin: links + // other OSes: helpful when dealing with payment-request files (in the future) was added to src/qt/paymentserver.cpp	N	src/qt/paymentserver.cpp		
f2bc1d35d6900e044209e76533dc514d31f7f8fe	# Note: Might need these options in the future for 64bit builds. was added to contrib/gitian-descriptors/boost-win32.yml	N	contrib/gitian-descriptors/boost-win32.yml		
35b8af92265ed74de63c3818e5290c27b3f35df2	# It can be turned off with MINGW_THREAD_BUGFIX=0, just in case it causes - # any problems on some untested qmake profile now or in the future. was removed from bitcoin-qt.pro[DF]	N	bitcoin-qt.pro		
2341e9b533fedf9bade5b55d676b2c0cf5ed8604	The string parameter is just an arbitrary name given to the default wallet. It's been prepended with a tilde to avoid name -collisions in the future with additional wallets. was removed and added back to doc/multiwallet-qt.md	N	doc/multiwallet-qt.md		
8f6709076ecae70e970ae3baae215bf67e5a4e5c	However, this release includes a bug -fix that makes it a little bit more difficult for attackers to double-spend a -certain type ("lockTime in the future") of zero-confirmation transaction. was removed from doc/release-notes.txt	N	doc/release-notes.txt		
c25e98186d0f716451ef000e55646d25e014f573	More types may be added in the future. was added to doc/log_format.txt [NF]. More meta block types +may be added in the future. was added to doc/table_format.txt[NF] , // We intentionally allow extra stuff in index_value so that we + // can add more features in the future. was added to table/table.cc[NF]	N	doc/log_format.txt doc/table_format.txt table/table.cc		
64753f0438f66c51ba7ae2173120ae97f22d186e	More types may be added in the future. was removed from src/leveldb/doc/log_format.txt[DF]. More meta block types -may be added in the future. was removed from src/leveldb/doc/table_format.txt[DF]. // We intentionally allow extra stuff in index_value so that we - // can add more features in the future. was removed from src/leveldb/table/table.cc[DF]	N	src/leveldb/doc/log_format.txt src/leveldb/table/table.cc src/leveldb/doc/table_format.txt		
67155d9299ef75cc73272a259dbfbf72632c3020	The string parameter is just an arbitrary name given to the default wallet. It's been prepended with a tilde to avoid name +collisions in the future with additional wallets. was added to multiwallet-qt.txt[NF]	N	multiwallet-qt.txt		
d38c6488d067c2e88726e2ca99bc76fd67dab49b	However, this release includes a bug +fix that makes it a little bit more difficult for attackers to double-spend a +certain type ("lockTime in the future") of zero-confirmation transaction. was added to doc/release-notes.txt	N	doc/release-notes.txt		
ef3988ca369900206b0cfc32cc1958aee0e43710	return error("CheckBlock() : block timestamp too far in the future"); was changed to return state.Invalid(error("CheckBlock() : block timestamp too far in the future")); in src/main.cpp	N	src/main.cpp		
5e650d6d2dbfc284c300668e71188e663d8f0a45	More types may be added in the future. was added to src/leveldb/doc/log_format.txt[NF]. More meta block types -may be added in the future. was added to src/leveldb/doc/table_format.txt[NF]. // We intentionally allow extra stuff in index_value so that we - // can add more features in the future. was added to src/leveldb/table/table.cc[NF]	N	src/leveldb/doc/log_format.txt src/leveldb/table/table.cc src/leveldb/doc/table_format.txt		
49e1501b5dbccdea03561684cc5aeccfac4e5e5a	# It can be turned off with MINGW_THREAD_BUGFIX=0, just in case it causes + # any problems on some untested qmake profile now or in the future. was added to bitcoin-qt.pro	N	bitcoin-qt.pro		
af836ad58808a1f6437a8b9a76b670c07c9a1212	// Last date that can be represented (far in the future) was changed to /** Last date that can be represented (far in the future) */ in src/qt/transactionfilterproxy.	N	src/qt/transactionfilterproxy.		

	ceb6d4e11d8dab8a6778e20c433f6ed989c16221	// Last date that can be represented (far in the future) was added to src/qt/transactionfilterproxy.cpp[NF] and src/qt/transactionfilterproxy.h[NF]	N	src/qt/transactionfilterproxy.cpp src/qt/transactionfilterproxy.h			
	69d605f410c83235aa7b757445e7d0166fcfe2d9	return error("CheckBlock() : block timestamp too far in the future"); was added to core/src/main.cpp[NF]	N	core/src/main.cpp			
	0a61b0df1224a5470bccdab302bc199ca5a9e356	return error("CheckBlock() : block timestamp too far in the future"); was removed from and added back to main.cpp	N	main.cpp			
	8bd66202c324a6c7a79abc0f1f0558dacbc59460	return error("CheckBlock() : block timestamp too far in the future"); was removed from main.cpp. and added to main.cpp	N	main.cpp			
	e071a3f6c06f41068ad17134189a4ac3073ef76b	return error("CheckBlock() : block timestamp too far in the future"); was added to main.cpp[NF]	N	main.cpp			NOTE [15]
	4405b78d6059e536c36974088a8ed4d9f0f29898	return error("CheckBlock() : block timestamp too far in the future"); was added to main.cpp[NF]	N	main.cpp			
will go away							
should be avoided	146746bbafe5dfea7b2019231c2d37fe57a4c3bb	In general, patches should be avoided. When a packager feels a patch is +necessary, the packager should bring the problem to the attention of the bitcoin +developers so that an official fix to the issue can make it into the next +release. was added to contrib/rpm/README.md[NF]	N	contrib/rpm/README.md			
TYPE1							
this isn't very solid							
don't use this	928b950e3b0622cca48a9eb2106dc7c006fbe520	// Another convenience method: call f approximately + // every deltaSeconds forever, starting deltaSeconds from now. + // To be more precise: every time f is finished, it + // is rescheduled to run deltaSeconds later. If you + // need more accurate scheduling, don't use this method. was added to src/scheduler.h[NF]	Y	src/scheduler.h			
get rid of this	e3bc56986508b40f894c7c707d39ea1bb437ebb5	// TODO: get rid of this .c_str() by implementing SecureString::operator=(std::string) was removed from src/bitcoinrpc.cpp and added to src/rpcwallet.cpp[NF]	Y	src/bitcoinrpc.cpp src/rpcwallet.cpp			
	94f778bdebe1438fbe0206621ce44b4e25196e59	// TODO: get rid of this .c_str() by implementing SecureString::operator=(std::string) was added to src/bitcoinrpc.cpp and src/qt/askpassphrasedialog.cpp	Y	src/bitcoinrpc.cpp src/qt/askpassphrasedialog.cpp			
	6ccff2cbdebca38e4913b679784a4865edfbb12a	#pragma warning(disable: 4700) // VC60 workaround: don't know how to get rid of this warning was removed from src/cryptopp/secblock.h[DF]		src/cryptopp/secblock.h	52e9fe929615d339732c060573b0637c51e353dc	--	
	6644d98d9eeaa24c197d18665b1461b8cd4300ea	#pragma warning(disable: 4700) // VC60 workaround: don't know how to get rid of this warning was added to cryptopp/include/cryptopp/secblock.h[NF]	Y	cryptopp/include/cryptopp/secblock.h			
	0a61b0df1224a5470bccdab302bc199ca5a9e356	#pragma warning(disable: 4700) // VC60 workaround: don't know how to get rid of this warning was removed from and added back to cryptopp/secblock.h	Y	cryptopp/secblock.h			
	3dd20ff2f87958a440c9dceae8abe3967aa48fcf	#pragma warning(disable: 4700) // VC60 workaround: don't know how to get rid of this warning was added to cryptopp/secblock.h[NF]	Y	cryptopp/secblock.h			NOTE [16]
	813505cc1313b7e191b787f93e573acb91fa1464	#pragma warning(disable: 4700) // VC60 workaround: don't know how to get rid of this warning was added to cryptopp/secblock.h[NF]	Y	cryptopp/secblock.h			
just abandon it							
fix this crap							

it doesn't work yet						
crap	85ad31ede7bc338079c8ae643542fde7ad83ce55	READ_STATUS_INVALID, // Invalid object, peer is sending bogus crap was added to src/blockencodings.h[NF]	N	src/blockencodings.h		
	5d743099b5fe77ba423110bea4f5dfd854fef3b2	// Make sure adding crap to the scriptSigs makes them non-standard: was removed from src/test/script_P2SH_tests.cpp	N	src/test/script_P2SH_tests.cpp		
	cf04d83624e2ca66d094b6026e48bdf86693d7de	// Seed random number generator with screen scrape and other hardware sources was removed from src/util.cpp[NF]	N	src/util.cpp		
	39f0d9686095bce469dbfa5233331a5d15c6545	// Make sure adding crap to the scriptSigs makes them non-standard: was added to src/test/script_P2SH_tests.cpp	N	src/test/script_P2SH_tests.cpp		
	2097c09a9b5dda5bfa0de1906a8c155c3c3b529b	// Seed random number generator with screen scrape and other hardware sources was added to core/src/util.cpp[NF]	N	core/src/util.cpp		
	0a61b0df1224a5470bccdab302bc199ca5a9e356	// Seed random number generator with screen scrape and other hardware sources was removed from and added back to util.cpp	N	util.cpp		
	8bd66202c324a6c7a79abc0f1f0558dacbc59460	// Seed random number generator with screen scrape and other hardware sources was removed from and added back to util.cpp	N	util.cpp		
	e071a3f6c06f41068ad17134189a4ac3073ef76b	// Seed random number generator with screen scrape and other hardware sources was added to util.cpp[NF]	N	util.cpp		NOTE [17]
	4405b78d6059e536c36974088a8ed4d9f0f29898	// Seed random number generator with screen scrape and other hardware sources was added to util.cpp[NF]	N	util.cpp		
inconsistency	5262fde0ecd19f1febbfcd488f2be41c5dff047	return error("VerifyDB() : *** irrecoverable inconsistency in block data at %d, hash=%s", pindex->nHeight, pindex->GetBlockHash().ToString()); was removed from and added to src/main.cpp	N	src/main.cpp		
	7d9d134bf95cb6a2ce9623c7e6a3535432a61af2	return error("VerifyDB() : *** irrecoverable inconsistency in block data at %d, hash=%s", pindex->nHeight, pindex->GetBlockHash().ToString().c_str()); was removed from and added to src/main.cpp	N	src/main.cpp		
	f8b7aa862519ab2efd1ce327d2ed4bea1325dc11	// No network rules currently depend on the version here, so an inconsistency is harmless + // but it must be corrected before txout nversion ever influences a network rule. was added to src/main.cpp	Y	src/main.cpp		
	6e51b3bddf782f53527cf968445b298ebdec9bbc	strErr = "Error reading wallet database: CPrivKey pubkey inconsistency"; was removed from src/walletdb.cpp	N	src/walletdb.cpp		
	dfa23b94c24aae6466152fccbe896ba5dc0e97b4	strErr = "Error reading wallet database: CPrivKey pubkey inconsistency"; AND strErr = "Error reading wallet database: CWalletKey pubkey inconsistency"; were removed AND strErr = "Error reading wallet database: CPrivKey pubkey inconsistency"; were added to src/walletdb.cpp	N	src/walletdb.cpp		
	1f355b66cd90f1bd96a862604a7216e8ba56f853	return error("VerifyDB() : *** irrecoverable inconsistency in block data at %d, hash=%s", pindex->nHeight, pindex->GetBlockHash().ToString().c_str()); was added to src/main.cpp	N	src/main.cpp		
	eed1785f701be93ac2464e854c2a7de1f748ef84	strErr = "Error reading wallet database: CPrivKey pubkey inconsistency"; AND strErr = "Error reading wallet database: CWalletKey pubkey inconsistency"; were added to and removed from src/walletdb.cpp	N	src/walletdb.cpp		
	9eace6b1130ce7eb938476750159ec0baf752531	printf("Error reading wallet database: CPrivKey pubkey inconsistency\n"); AND printf("Error reading wallet database: CWalletKey pubkey inconsistency\n"); were removed from src/db.cpp and added to src/walletdb.cpp[NF]	N	src/db.cpp src/walletdb.cpp		
	3cc06249322bac5c495848a3f97be508153dc5d9	printf("Error reading wallet database: CPrivKey pubkey inconsistency\n"); AND printf("Error reading wallet database: CWalletKey pubkey inconsistency\n"); were added to src/db.cpp	N	src/db.cpp		
take care	e4f126a7ba66e7317718c30276dff6db92dc1986	// We don't hold a direct pointer to the splash screen after creation, but the splash + // screen will take care of deleting itself when slotFinish happens. was added to src/qt/bitcoin.cpp	N	src/qt/bitcoin.cpp		
	695041e4952ea40e02e3b0431d04ca6af7ec9711	// Take care not to overread C strings in truncating conversions like + // "%.4s" where at most 4 characters may be read. was added to src/tinyformat.h	N	src/tinyformat.h		

	75a880390191aeaf7d2fa326f194349a891db022	Making use of this, third-party scripts can be written to take care of +transaction (re)broadcast: was added to doc/release-notes/release-notes-0.11.0.md[NF]	N	doc/release-notes/release-notes-0.11.0.md			
	3e8ccc765a6d02b701ebc40e556e32e4c0613987	Making use of this, third-party scripts can be written to take care of +transaction (re)broadcast: was removed from doc/release-notes.md	N	doc/release-notes.md			
	b679a6b4038d12aa9c194c777b1f91d71cb076a4	Making use of this, third-party scripts can be written to take care of +transaction (re)broadcast: was added to doc/release-notes.md	N	doc/release-notes.md			
	023e63df78b847812040bf6958c97476606dfbfd	// Miscellaneous updates -- nothing to do, status update will take care of this, and is only computed for visible transactions. was removed from and added back to src/qt/transactiontablemodel.cpp	N	src/qt/transactiontablemodel.cpp			
	3015e0bca6bc2cb8beb747873fdf7b80e74d679f	// Miscellaneous updates -- nothing to do, status update will take care of this, and is only computed for + // visible transactions. was added to src/qt/transactiontablemodel.cpp	N	src/qt/transactiontablemodel.cpp			
	7149499fd85d5adea23c9c3057944c3f2f69a2d2	// 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. was added to src/qt/bitcoinunits.cpp	N	src/qt/bitcoinunits.cpp			
	ada5a067c75f19a724cc054286ecf2254e5dbef	// Miscellaneous updates -- nothing to do, status update will take care of this, and is only computed for - // visible transactions. was removed from /src/qt/transactiontablemodel.cpp	N	/src/qt/transactiontablemodel.cpp			
	fe4a655042f7de31dce120aedd72345579f1b59f	// Updated -- nothing to do, status update will take care of this was removed from src/qt/transactiontablemodel.cpp AND // Miscellaneous updates -- nothing to do, status update will take care of this, and is only computed for + // visible transactions. was also added to it.	N	src/qt/transactiontablemodel.cpp			
	0f3981bea94cea957f0de4b128f7feffbc2d9c6	/* Updated -- nothing to do, status update will take care of this */ was changed to // Updated -- nothing to do, status update will take care of this in src/qt/transactiontablemodel.cpp	N	src/qt/transactiontablemodel.cpp			
	66d536ed0765c6369738c2b35dd528dac3f5ee67	/* Updated -- nothing to do, status update will take care of this */ was added to gui/src/transactiontablemodel.cpp	N	gui/src/transactiontablemodel.cpp			
something's gone wrong							
unknown why we ever experience this							
treat this as a soft error							
this isn't quite right							
this doesn't look right							
is this line really safe							
impact of the past design							

impact of the past decision							
quick fix							
better scheme needed							
You have been warned							
a really bad idea							

[1] Commit messages almost similar

[2] Commit messages almost similar

[3] Commit messages almost similar

[4] Commit messages almost similar

[5] Commit messages almost similar

[6] Commit messages almost similar

[7] Commit messages almost similar

[8] Commit messages almost similar

[9] Commit messages are similar

[10] Commit messages are similar

[11] Commit messages are similar

[12] Commit messages are similar

[13] Commit messages are similar

[14] Commit messages are similar

[15] Commit messages are similar

[16] Commit messages are similar

[17] Commit messages are similar