QC™ Release Notes

This document contains version history for QC™ since the first shipping (1.0) version. If you have questions about this document, please contact Onyx at one of the addresses below.

• **v1.5 (PPC) 8/30/2000**
- **New Features/Changes**
- QC is now compatible with Mac OS 9 and Carbon applications.
- QC fixes some bugs while testing with VM on, and properly identifies (e.g. doesn't report errors on) Font metric data stored outside of an application heap by the system.
- QC v1.5 is for use on PowerPC Macintosh systems only. Testing of 68k code running emulated on a PPC Mac is still supported.
- **Documentation Changes**
- The BadAPPL sample application has been updated to build with CodeWarrior Pro 5.

• **v1.2.5 (PPC) 1/1/1997**
• **v1.1.4 (68K) 1/1/1997**
- **New Features/Changes**
- Pascal interfaces to the QCAPI have been added to the 68K version.
- The MPW make files for BadAPPL have been updated to support xcoff usage and PowerPC builds.
- The QCPBRec parameter block for QCAPI callbacks has had the 'errData' array of 4 longs added to it. The contents of this array can vary for each reported error but will contain address/handle/ptr data for pertinant errors like Handle usage errors and block moves. For example, if the error is 'Handle does not point to reloc block: 001734F8', errData[0] will contain the '001734F8' value. This modification has been made for third party debuggers trying to get at QC reported error information.
- **Documentation Changes**
- The QCAPI Manual has been updated for the QCPBRec addition noted above.

• **v1.2.4 10/10/1996**
- **New Features/Changes**
- QC handle and pointer validation routines have been modified to ignore errors in code not targeted for testing. This helps avoid common errors like 'NIL Handle Warnings' caused by NIL handle usage in the OS or third party extensions.
- **Documentation Changes**
- The QCAPI Manual has been updated to properly document the result code of QCDeactivate() as being of type QCErr. Previous documentation showed it as returning a result of type Boolean.

• **v1.2.3 08/16/1996**
- **Bug Fixes**
- Corrected a bug that would sometimes crash the machine when attempting to play the activate sound during an auto-launch with InitGraf and no other sounds had been played on the Mac yet.
- Corrected a bug in removing block bounds tags that could occur when quitting an application being tested, without deactivating QC first, and then relaunching the same application. The relaunch of the application would sometimes crash.
- Corrected a bug in calling error handlers that would somestimes result in a crash after an error handler was called. This problem was most commonly seen when using the CodeWarrior's MW Debug and testing with QC.
- The QCAPI.h interface file now has alignment options to keep 68K alignment on for the structs. This could cause problems in PPC code that uses the QCAPI but compiled with PPC alignment turned on instead of 68K alignment.

- The BadAPPL sample application has had a few bugs fixed in it and updated to version 1.2.2.

• **v1.2.2 04/18/1996**
- **Bug Fixes**
- Corrected a problem that was found in the way QC reports the PC address of the cause of an error. The bug was unfortunately introduced in between 1.2 and 1.2.1 due to a development environment change. It was reporting 'PC of cause not found in target!' most of the time.

• **v1.2.1 03/21/1996**
- **Bug Fixes**
- An incompatibility with Open Transport that sometimes caused crashes during startup has been corrected.
- Corrected a bug in a PurgeMem patch that affected memory usage and sometime caused crashes.
- Added the value '0xFFC10000' to the "Zero Ok" (ZrOk) resource. This is a common value that gets written to address zero on 9500's so QC will ignore it now by default.

- **Documentation Changes**
- The QCAPI Reference document has been updated to correct reflect the interfaces for QCGetTestState and QCSetTestState. The previous docs were different than the actual interfaces defined in QCAPI.h.
- The QCAPI Reference document has been updated to better define how the QCHeapCheckNow, QCHeapScrambleNow, and QCHeapPurgeNow tests operate.

• **v1.2 1/22/1996**
- **New Features/Changes**
- QC 1.2 implements QC testing functionality in a native PowerPC environment for PowerMacs. Version 1.2 is for PowerPC Macintosh users only. A later version will be 'FAT' and contain both 68k and PPC testing code.
- The 'Block Bounds Checking' and 'Invalidate Free Memory' tests are now Modern Memory compatible. These tests are now available whether MM is on or off.
- Options for 'Debugger Breaks' have been added giving the ability for you to select whether you want QC to report errors with DebugStr() or SysBreakStr(), depending on the debugger you are using.
- Options for 'Validate Handle/Ptrs' have been added. The user can now choose whether reporting usage of NIL Handles or Pointers is active or not. We've had quite a few users request the ability to ignore 'Nil Handle Error' reports and this option now provides that ability.
- Renamed 'MemErr Detection' as 'MemErr Warnings' to better reflect what the test does.
- Changed the name of the qcapi routine 'QCDeActivate' to 'QCDeactivate'. Note the absence of the capital 'A' in 'activate'. A macro exists under the old name for compatibility.
- New activate and deactivate sounds have been added. These sounds are not specific to English speaking countries like the previous 'On'/'Off' spoken word sounds were.
- The QCAPI now supports Pascal calling conventions so Pascal projects can use the QCAPI.

- **Bug Fixes**
- Many tests include substantial speed improvements over the previous version due to better algorithms and optimizations.
- The error 'kBadBlockLenErr' generated during a Heap Check if a block length failure occurs, has been enhanced to report the block address as well as the length. Previously, only the length was reported.
- Implemented new hot key detection code with better support for international keyboards.
- Fixed a bug in launch list handling that sometimes resulted in duplicated items appearing checked (on) when QC was active and the Control panel was opened.
- QC no longer reports 'Nil Handle Error' that sometimes occurred during DrawPicture calls. Depending on the PicHandle being displayed and if it had NIL PicComments in it.
- When QC reports MemErr related warnings, the mem error value is now correctly displayed (i.e. -117 instead of 0xFFFFFF8B) in the output.
- Corrected a networking/socket error that caused debug versions of OpenTransport to user

break (DebugStr) and give a warning during system startup. The system would boot and operate successfully.

- **QCAPI Changes**

You can use v1.2 QCAPI libraries with previous versions of QC as well as use the v1.1.3 QCAPI libraries with QC v1.2. They are equally backward and forward compatible. However, some new items have been added that will result in error results when used with an older version of QC (pre 1.2). Previous versions will not understand the parameters passed in calls and return an error.

- You can now set 'Debugger Break' preferences (DebugStr or SysBreakStr) with the QCSetTestOptions() api call. The QCTestOptionsRec has been expanded to accomodate debugger options.
- API control over 'Nil Handle/Ptr Warnings' is now available via the get/set api options (as mentioned above).
- All QCAPI routines are now declared Pascal for compatibility with C and Pascal development environments.
- If you are installing a Pascal error handler, use QCInstallPascalHandler() instead of the QCInstallHandler() routine intended for C handlers. Removal of a C or Pascal error handler is still done with QCRemoveHandler().

- **QCAPI Compatibility**

Any application that was built with a previous QCAPI library (i.e. QC 1.1.3) will still work with QC 1.2 installed. Likewise, an application built with QC 1.2 api libraries will still work if QC 1.1.3 is installed on the Mac. If QC 1.1.3 or earlier is installed, calls to set 'Debugger Options' or 'Nil Handle/Ptr Warnings' with QCGetTestOptions()/QCSetTestOptions() will fail with a 'kQCInvalidType' result code. This is because these options were not a feature in QC 1.1.3 or earlier.

• **v1.1.3 04/21/1995**

- Fixed a crash that occurred when the sound volume was set to zero and QC was activating/deactivating.
- Fixed a problem where 'es'ing or force-quitting an app would leave QC in a state where it couldn't be activated with the hot keys.
- Fixed a bug in the cdev that would prohibit trying to add a new file to the launch list when an existing file in the list had the same name and type OR creator (but not both type AND creator). QC would report that it already existed in the list and not let the user add it. An example would be trying to add a file called 'SampleApp' with type/creator 'APPL'/'SAMP', when a file called 'SampleApp' with type/creator 'appe'/'SAMP' already existed in the list.
- Fixed a visual problem in the Block Bounds Options dialog in the control panel that would display the tag size dimmed on 1-bit deep color ports. This was most commonly seen on a non-color PowerBook.
- Added separate sounds for error/activate/deactivate at the request of our users.
- PowerBooks entering sleep mode call the Power manager which overwrites location zero. QC has been modified to temporarily turn off location zero testing (if those tests are on) while a PowerBook sleeps.
- The prefix 'QC:' now appears before error reported by QC with DebugStr(). The error string given to an api error handler does not contain the prefix.
- QC was incorrectly purging purgeable handles that were the target of SetHandleSize calls. When the SetHandleSize call increased their size QC would purge them even though they were the handles being resized. This has now been fixed.
- Set up a true QD environment so that spurious patches that attempt to use QuickDraw on whatever A5 value happens to be won't bomb.
- The 'BlockMove Checking' test will now ignore an error of the source pointer being NIL when the StandardFile package is in use. Previously, users got a QC user break reporting that the source pointer was NIL when StandardGetFile()/StandardPutFile() calls were made. Since this is in the Standard File Package, users can't do anything about it anyway.
- Added some resources for easier localization of text displayed in various dialogs and list displays.
- Added the Think Project Manager and Symantec Project Manager applications to the ignore

list ('Ignr' rsrc) so QC doesn't autolaunch on them when they are building resources that QC is supposed to auto-launch on. For example, setting up auto-launching on a 'cdev' resource to test a control panel would trigger QC when TPM or SPM built that 'cdev' resource.
- Added Metrowerks 68k & PPC Compiler file type/creator signatures to the 'Ignr' list.
- Added Resorcerer file type/creator to the 'Ignr' list.

- **QCAPI Changes**
The following are descriptions of changes to the QCAPI, including some interface changes. Applications that use the QCAPI will need to be rebuilt with the appropriate v1.1.3 QCAPI library.

- Added the new QCHeapCheckNow() call. What it does should be obvious.
- Multiple error handlers can now be installed across process boundaries. That is, multiple applications can all install error handlers at the same time. Unlike system trap patches, QC handles the calling of multiple error handlers eliminating the need for each handler to call the previous handler.
- Created a QCCallBackUPP and macros for creating MixedMode routine descriptors for PowerPC code installing an error handler. This ensures that the 68k INIT code will be able to call it ok.
- The QCAPI routine QCInstalled() now returns a QCErr instead of a Boolean. This routine will return 'kQCNoErr' if QC is installed (e.g. the routine didn't fail), and can now return the new 'kQCAPIMismatch' result along with previous declared possibilities.
- Modified QCAPI to return 'kQCAPIMismatch' when the loaded extension code does not match the QCAPI library in use. This helps diagnose problems better then the [previous] result of 'kQCNotInstalled'.
- The QCInstallHandler() routine can now possibly return the error 'kQCPBRecMismatch' if the version of the QCAPI being used is different than what the version of the extension code expects.
- Installed QC error handlers now remain installed across activate/deactivate calls. BadAPPL has been rev'd to install a handler once now instead of in each test to illustrate this.
- Added 'lastTestTrapPC' and 'lastTrapPC' to QCPBRec param block for error handlers installed with QCInstallHandler(). Third party debuggers can now use this address to map things back and display source code for where there error was detected.
- There is now a PowerPC shared library called 'QCAPI Lib' that can be used in PowerPC based projects. By using a shared library, it can be used in CodeWarrior, MPW, and Symantec environments.

• **v1.1.2 09/01/1994**
- Fixed a bug that occasionally caused double disposes of a Pointers to not be detected. Bug #59
- Fixed bug that occured when using QC and the Think debugger. QC would deactivate after some errors were reported in the Think debugger. Bug #50
- Debug break messages are now multi-line and more descriptive.
- PowerPC versions of the QCAPI libraries can now be used. QCAPI.c has been modified to use Mixed Mode to call the 68K INIT code. CodeWarrior only for now. Since the native PowerPC QC is still in the works this is just a temporary library available for those trying to use the API from PPC code.
- A BlockMove to a NIL destination or from a NIL source is now reported with more detailed text indicating where the error was detected. Previously the error was simply that a NIL pointer was used but no indication of what caused it.
- Adding an item to the launch list now automatically selects the newly added item. The most common operation after adding an item is to edit the options for it so auto-selecting the added item allows you to just press ENTER to edit it.
- QCAPI Reference Manual documentation has been correctly modified to show that QCInstallHandler() takes two parameters and not just one. The QCAPI.h file had it right, the documentation was just wrong.
- QCAPI library now includes it's own string copy routine not requiring including projects to link with an ANSI library.
- Fixed bug in the QCAPI routines that would let them be used even if QC was disabled

during startup by either the option key (user disabled it) or QC couldn't load due to some error. Using the API calls when QC was not completely loaded like this would result in a definite crash.

• **v1.1.1 06/01/1994**
- A bug has been fixed in the Block Bounds Options dialog that incorrectly defaulted to a tag size of 4 when 64 was selected. Bug #55
- QCAPI libraries for C++ projects are now provided.

- **Preference Changes**
Version 1.1 includes some preferences changes. Upgrading to v1.1 will automatically throw away any preferences you have in your 'QCPrefs' file (in your Preferences folder). There is no need for you to throw away this file before upgrading.

• **v1.1 - 05/15/1994**
- Faster System Heap scrambles.
- Added options dialog for 'Block Bounds Checking' in test list. This dlg allows the user to specify the tag size (up to 64 bytes!) to be used in the block bounds checking logic.
- Support for single stepping in the THINK 'C' debugger, which didn't work before because the debugged code is being executed out of the THINK debugger heap.
- Added API call QCBlockBoundChecksNow() to perform a block bounds check immediately instead of having to wait on the next memory manager call.
- The existing API calls QCSetTestOptions() and QCGetTestOptions() can set/access the block bounds tag size by using the new 'QCBlockBoundsRec' struct in the 'QCTestOptions' union. See QCAPI.h for the details.
- Fixed system heap scramble bug that could cause boundary overwrites.
- Fixed error that would occur on 68000 machines if a block bounds error was detected.
- Corrected ZapMem() bug that caused incorrect values to be used to invalidate blocks (instead of 0x52ff8001) on 040 Macs.Bug #46
- Fixed bug with free mem invalidation during MoveHHi on 24 bit heaps that would cause the block being moved to be invalidated. Bug #49
- Corrected a bug that would sometime disable QC after reporting a block overwrite error. This most commonly happened when running with the Think Debugger. Bug #50
- Correctly bitmapped the word 'DEMO' displayed along with the QC logo in demo versions of QC. Bug #48
- Fixed system heap scramble bug that could cause boundary overwrites.

• **v1.0 04/12/1994**
- First shipping public release

Onyx Technology
7811 27 Ave W.
Bradenton, FL 34209
Main: 941.795.7801
Fax: 941.795.5901

AOL: OnyxTech
AppleLink: D2238
Internet: support@onyx-tech.com
ftp://ftp.onyx-tech.com/pub/QC/
http://www.onyx-tech.com/
Mailing list: qc-info@world.std.com
To subscribe send 'subscribe qc-info your@address.goes.here' in msg content to the address 'qc-info-request@world.std.com').