Notes on Version 5.00 (Commercial)

- Another major redesign. All earlier versions of QSORT found the sort keys in-place when comparing records. Version 5 extracts keys during pre-processing and builds a fixed-length key block. This yields a major performance improvement, particularly when using multiple keys. The down side is that the length of the "sort record" is increased. If merge files are created, they use more disk space. Of course, the key block disappears during final output.
- Added several new key types not feasible without key extraction, among them numeric keys (ASCII), and three binary numeric key types.
- Added a whole new program, dQSORT, to sort data bases (one beta tester reported about 20% faster than Clipper). It will only be available in the commercial package.

Notes on Version 4.11

Fixed two bugs:

- Under certain circumstances, inverting one key would invert following keys.
- When sorting records with a variable number of delimited fields, missing keys should sort low. They were, in fact, sorting in arbitrary order and causing strange results.

Notes on Version 4.10

- Updated documentation to reflect the fact that version 4 is last shareware version. Except for salutation, code is unchanged.

Notes on Version 4.04

- Corrected a bug in merge output routine. When sorting large file with one-character line terminator, program got stuck in a loop and filled up disk.

Notes on Version 4.03

- Some minor cosmetic changes in the help screen. Otherwise identical to version 4.02.
- This product is now released as shareware.

Notes on Version 4.02 (Commercial)

- Corrected a bug which caused incorrect sorting of files of delimited records.

Notes on Version 4.01 (Commercial)

- Corrected a minor bug in the handling of the delimiter character in delimited field records.

* Moved all "natural language dependant" strings and messages to a special include file to facilitate translation to other languages.

Notes on Version 4.00 (Commercial)

* Massive rewrite of QSORT internal routines to enhance performance, particularly when sorting large files. QSORT now uses all of available memory as its sort buffer, so the more memory available, the better it works. Tests on my machine indicate a 2:1 performance improvement doing a simple sort of a 4.5 meg file with a 400K+ sort buffer available. One beta tester showed a 6:1 performance improvement sorting a 100 meg file!
* Added provisions to prevent merge file name collisions when multiple copies of QSORT are run concurrently on a network.
* Added /N<term>[<term2>] parameter for newline redefinition.
* Removed newline redefinition from /D parameter, and relaxed requirements for specifying number of fields. If the delimiter is not newline, <fields> need not be given, and QSORT will permit a variable number of fields in each record.
* Added @<command_file> parameter to permit QSORT parameters to be stored in an external file, and used repeatedly.

Notes on Version 3.22

* Documentation changed to reflect new shareware policy, and to point out the need to enclose the messages redirection parameter ("2><file>") in quotes for later DOS versions.
* No program changes.

Notes on Version 3.21

* FINALLY fixed the bug causing QSORT to not notice it had run out or disk space! (Thanks to Andy Woolls for reporting the problem still existed.)
* Added standard error redirection feature (see QSORT.PRN).
* Modified command line parameter parsing. You may now reliably run parameters together (not recommended, but you can do it). However, now a slant bar (/) may only be used as a tag character in a /T parameter, or a delimiter or record terminator character in a /D parameter if it is preceded by a back-slant (\). Otherwise, a slant bar marks the beginning of a new parameter. The caution on this subject which appeared in 3.20's notes has been removed, as it no longer applies.

Notes on Version 3.20

* This started out to be maintenance release 3.14, but so many changes were made, it certainly warrants a minor release number.
* Added a subparameter (V) to the /S parameter. /SV causes running status reports to be displayed on the console.

* Corrected a minor bug in the statistics display. When you sort a file with more than 32767 records, the number of records reported looked strange.
* Revised the method of displaying error messages, and made error message text more or less consistent.
* Added a whole new section to the documentation dealing with error messages and their causes.
* Revised the method of parsing the command line. Contrary to the documentation, you may now run command line switches together. "/S/R/?/20:6/15:5" will do precisely what you would expect.

Notes on Version 3.13

* This release cleans up the last (I hope) potential undetected I/O errors.

Notes on Version 3.12

* A bug (an oversight, actually) has existed for some time which caused QSORT to fail to recognize an "out of disk space" condition when writing temporary merge files. The situation is corrected in this version.
* The QSTMP environment variable was documented in version 3.10, but through another oversight, not actually implimented 'till now. The current version will search for environment variables QSTMP, TMP and TEMP in that order, looking for a place to put temporaries.

Notes on Version 3.11

* Rewrote many internal functions in assembly language.

Notes on Version 3.10

* Added support for delimited field records. A side benefit of this feature is the ability to redefine the newline sequence for this and other types of ASCII files.
* Improved output buffering.
* Converted all documentation to Microsoft Word.

Notes on Version 3.00

* Converted from Lattice C, version 3.1 to Microsoft C, version 4.0. This change alone resulted in about a 10 % improvement in performance.
* Performed major redesign of QSORT's I/O, buffer handling and internal data structures. This change resulted in an additional 60 % performance improvement. This improvement was measured by sorting the same 210K file:

```
   QSORT 2.1   1:48
   QSORT 3.00  0:42
```

* Dropped support for QSORTL (large data model).

* Removed the necessity for the /M<len> parameter. QSORT will now sort files with variable length records of any size, provided only that the two largest records will fit together in the sort buffer. This imposes a record length constraint of about 50K for the sum of these record lengths.
* Added the option of specifying the output file as a command line argument. The following two commands are exactly equivalent:

```
   QSORT <INFILE >OUTFILE      and
   QSORT  INFILE  OUTFILE
```

* Added support for the /? help parameter.
* Added the /S parameter to produce a "statistics" screen report at the end of a QSORT run.
* Added support for the QSTMP and TMP environment variables to specify a path for temporary merge files.
* QSORT documentation was largely rewritten.

Notes on Version 2.1

* Converted form Lattice C version 3.0 to version 3.1.
* Added /F<len> parameter and support for fixed-length records. Records may contain binary data, but keys must be ASCII.
* Added several special-purpose record compare routines. QSORT selects the most efficient one for a given sort, based upon the combination parameters on the command line.

Notes on Version 2.0

* Converted from Lattice C, version 2.14 to version 3.0. This change achieved about 15 % performance improvement.
* Began distributing both QSORT and QSORTL (large data model). The latter shows performance gains on very large (> 750K) files.
* Added /T[<tag_char>] parameter and support for multi-line logical records.

Notes on Version 1.2

* Added "lexical" ordering of key fields.

Notes on Version 1.1

* Added /M<len> parameter to permit specifying maximum record lengths in excess of the default 132 bytes.
* Corrected bug to permit <col> element of key field specifier to be optional, defaulting to column 1. Program now matches documentation.