ee9 Release History

V0.1, 04-Nov-2010: First alpha release

V0.2, 24-Nov-2010: Second alpha release

a. I advanced the version number to 0.2.

b. I fixed an inconsistency, between fast mode and other diagnostic modes, in the measurement of the KDF9 virtual CPU time consumed when running a program.

c. I improved the declarations of some of the fundamental types in `KDF9.ads`.

d. I provided a fail-safe way of dealing with programs that corrupt their `E0U`, so avoiding difficulties for postmortem diagnostic printing in `U` format.

e. I fixed a bug in the reconstruction of normalised floating-point results from separate fraction and exponent parts.

f. I added a facility to supply multiple 'rolls of tape' to a tape reader.

V0.3b, 20-Jan-2011: Third alpha release

a. I advanced the version number to 0.3, and the copyright date to 2011.

b. I added a facility to provide a pre-prepared set of replies to console Flexowriter prompts, with the facility to make conditional responses. (This nods at the edge-punched card reader of the console Flexowriter.) An audible warning is given (i.e., `BEL` characters are output) when manual Flexowriter input from the user is needed.

c. I enhanced the setting-file feature to provide separate settings for a first program (e.g., the Whetstone Translator) and overlay (e.g., the Controller).

d. I much reduced **ee9**'s verbosity by omitting most confirmatory messages, and added the option to suppress even the output of the final KDF9 state.

e. I fixed the timing of I/O transfers that are on-going at the end of a run, and of I/O transfers that are held up by store or buffer lockouts.

f. I fixed a major bug that was detected by Brian Wichmann in the '×D' order.

g. I fixed the implementation of '–D', which was adding the top 48 bits of its operands instead of subtracting them. This had gone quite unnoticed until I called it as a subroutine in the new implementation of '×D'!

h. I fixed the usage–error message from **ee9**, given in response to incorrect flag parameters.

i. I decoupled the generation of digital signatures from the running trace mode, thus making signature generation a more practical means of verifying correct operation by an **ee9** port or new version.

j. I unified the conditions under which retrospective tracing, digital signature generation, and running trace output, are all activated.

k. I improved the shell commands in `Testing`, to make running **ee9** easier, less error prone, and better diagnosed in the case of an error.

l. I provided a `HOWTO` file that gives more complete instruction on the operating procedures to be followed for compiling and running KDF9 programs.

m. I implemented a first cut at an automatic self-testing process to check the correct operation of an installation or new port of **ee9**.

n. I simplified the handling of the diagnostic mode options, both internally and *vis-a-vis* the user-interface semantics; the latter now make some sense.

o. For revision 0.3b I improved the installation and self-testing code, in line with feedback from David Holdsworth.

V0.4, 15-Feb-2011: Fourth alpha release

a. I advanced the version number to 0.4, and added a message to identify the version of **ee9** that is being used.

b. I improved the encapsulation of the `IO` package, by making `IO.stream` a private type.

c. I made `IO.stream` a tagged type, allowing prefix notation, so that component access could be re-interpreted as a call on a 'getter' function.

d. I tidied up the diagnostic-mode confirmation message.

e. I forced runtime checks in all modules, except some in the `kdf9` hierarchy; this greatly improves self-diagnosis, yet has no perceptible impact on the performance of an optimised build.

f. I tidied up the contents of the `Testing` directory by introducing the `Binary` subdirectory for KDF9 machine code programs, and by renaming `Usercode` as `Assembly`. This puts all the subdirectories together in a listing.

g. I created a command, **tsd**, to simplify running the **T**ime **S**haring **D**irector.

h. I arranged for the **tsd**, **nine**, and **whet** commands to set the `FW0` file suitably for their own type of program run, restoring it on termination to its former contents.

i. I tidied up `IOC` a little.

j. I moved digital signature generation into `kdf9`, where it properly belongs.

k. I factored mode testing, and logging, out of the `check_*_points` procedures, gaining an 8% speedup in fast mode and no slowdown in other modes.

l. I restricted the signature hashing to the two possibly-relevant Q stores, instead of all 16. This speeds up a run of **ee9_self_test** by a factor of 2.5!

m. I added a facility to include Line Shift and Page Change characters in Flexowriter prompts, using the Latin-1 characters ® and © respectively.

n. I implemented the (EE model 1081) `MT` orders, with a (hopefully temporary) restriction that blocks are limited to a maximum of 512 KDF9 words in length.

o. I corrected the timing of Flexowriter writes that include fillers and/or inject canned responses to prompts.

p. I implemented OUTs 4 and 10, to let problem programs such as Whetstone Algol claim magnetic tapes as they would when running under Director.

q. I made a version of the Whetstone Benchmark, called `MTW`, that outputs to a `MT` labelled `WHETLIST`, instead of OUT 8 stream 30.

r. I included four `MT`-handling programs in the self-testing procedure. They are `RLT` (Rewind and Label Tape); `TRB` (Test Read Backwards); `OUT4` (tests the new OUT); and `OUTX` (tests OUT 10). I also included `MTW`.

V0.5, 15-MAR-2011: FIRST BETA RELEASE

a. I advanced the version number to 0.5; this is the first beta release, the change in status being justified by the lack of bug reports from heavy users.

b. I fixed previously undetected bugs in '–DF' and '×+±*n*'.

c. I provided a lax nest-depth checking option (option `N`), to better approximate the behaviour of the KDF9 hardware when the nest is under-filled.

d. I fixed bugs in the nest-depth checking of some 1– syllable and 2–syllable orders.

e. I improved the promptness of time-limit expiry in fast mode: it is now put off to the next successful jump, rather than to the end of the time slice.

f. I improved the discrimination of the digital signature by hashing `ICR` into it.

g. I speeded **ee9** up by ~20% over V0.4: by judicious in-lining of store access primitives, by refactoring the instruction decoding, by removing some redundancy from several Q-store orders, and by implementing a better method of ensuring that Q0 is always 0; all this with no adverse effect on maintainability or modularity.

h. I replaced the make command as a build tool with a shell command file called `Build/mk9`, which takes the same parameter options as the `makefile` formerly did.

i. I added a section on building **ee9** to the `HOWTO` file; and included URLs for EE Director documentation and the EE KDF9 Algol Manual.

j. I made pop operations zeroize the affected nest cell(s), as the hardware did.

k. I changed the `V` option flags to a more mnemonically consistent set: `DHIJPS`.

V1.0V, 21-APR-2011: FIRST PUBLIC RELEASE

a. I advanced the version number to 1.0: **ee9** goes forth into the world!

b. I fixed previously undetected bugs in 'NEGD' and '–D'. Thanks to the latter, and to David Holdsworth's discovery of a mis-transcription in the Whetstone Controller's Usercode, Walgol numeric input now works properly.

c. I completely revamped the tracing feature, replacing the jump and skip traces with a single, consolidated retrospective trace of all instructions executed while tracing is enabled. Each traced instruction is accompanied by its most relevant operand (see the User Guide for details).

d. In conformity with (c), I changed the trace mode specifiers from `JUMP_MODE` and `SKIP_MODE` to `RETRO_MODE`; and the verbosity (option `V`) flags from `D` to `S`, and from `J`/`S` to `R`; again, with further improvement in mnemonic quality.

e. I speeded **ee9** up by ~10% over V0.5 by removing all tracing overhead from the fast mode code path. Tracing now invokes the fast mode code as a subroutine, wrapping it in calls that determine the operand to be traced. As well as being faster, this is more modular, since the details of tracing can now be changed without touching the fundamental emulation code.

f. I wrote a program, `Algol/input.a60` to test Algol numerical input; and included it in the self-testing procedure.

g. I added a verbosity option to the command-line parameters of **ee9** (see the *Users' Guide*), **whet**, **nine**, **dow**, and **tsd**. This allows the tracing verbosity to be set without needing a `V` line to be put into a settings file.

h. I changed the self-test procedure to use the new verbosity option, instead of setting up a `V` line in the settings files.

i. I reverted change (e) of V0.5, as its adverse effect on performance became more onerous as a result of V1.0's increases in overall efficiency.

j. I corrected the absence of a final `OUT 0` or `OUT 1` order from the tracing.

k. I simplified the self-test logs by omitting the 'last output line' feature.

l. I improved the layout of tracing output, reducing the screen-width needed to view it without wrapping lines.

m. I removed the facility to monitor locations changed by peripheral input.

n. I allowed for separate post-run dumps for a program and its overlay (e.g., the Whetstone Translator and Controller).

o. I added a warning to **ee9_self_test**, re: overwriting of I/O device files.

p. I rewrote the circular and double-length shift orders in conformity with EE engineering document K/GD.y.80, "KDF 9: SHIFTING AND SHIFT CONTROL".

q. I rewrote `KDF9.as_Q()` and `KDF9.as_word()` as normal functions, avoiding a portability problem revealed by the MacOS X PPC Leopard port (with thanks to Mike Hore).

r. I rewrote the way **ee9** ensures that Q0 is never set to a non-zero value. In `LAX_MODE` an assignment to Q0 is now suppressed; in `STRICT_MODE` it is now treated as an execution error.

V1.5W, 01-AUG-2011: SECOND PUBLIC RELEASE

a. I advanced the version number from 1.0v to 1.5w, because adding support for Microsoft Windows seems like quite a big step. I am mindful that LISP never got past its version 1.5, but hope that there will be V2 of **ee9**!

b. I corrected the instruction timings for =[R]{QCIM}q type orders; as a result the **ee9** timing for the Whetstone benchmark is more accurate: 420.7s, down from 422.7s, the real KDF9's measured time being about 417s.

c. I added a test, in trace mode cycles only, to verify that Q0 = Q0/0/0 at the end of each instruction execution.

d. I discovered that, with GNAT GPL 2011, optimised builds with all runtime checking suppressed run little (~6%) faster than with most checks enabled. Now the default **ee9** build is optimised, with runtime checking.

e. I added new verbosity flag options: `A` to suppress the output of messages when **ee9** services an OUT **A**PI successfully; `E` to suppress the output of mere progress messages from **ee9**; `F` to suppress the output of the KDF9's **F**inal state; and `Z` to suppress **all** output from **ee9** itself, rather than KDF9.

f. I amended the Flexowriter emulation package to use separate input and output file handles. This was necessitated by Microsoft Windows, but has been done in a way that is completely source-code compatible with UNIX systems.

g. I provided alternative bodies for `open_ui()` that opens the terminal user interface: one for Windows and another for UNIX systems.

h. I amended `IO.open()` to ensure that I/O transfers are transparent to all byte values. This is necessary on Windows to allow KDF9 binary machine code to be read uncorrupted, but has no effect on UNIX systems.

i. I recast the **mk9** shell command to better support builds of **ee9** for Windows.

j. I amended the shell files in `Testing` to eschew the use of file system links, which do not work well under Windows (to put it mildly).

k. I amended the way **ee9** ensures Q0 is never set to a non-zero value so that, in Director state, an assignment to Q0 is suppressed even in `STRICT_MODE`.

l. I provided separate subroutines for Windows and for others, to isolate the change of font colour in Flexowriter I/O. In this first version, the colour is **not** changed on Windows.

m. I amended `actual_prompt()` to cope with the presence and/or the absence of ANSI-terminal font colour escape sequences.

n. I consolidated the Windows amendments for Flexowriter output into a single package, `terminal`. Alternative bodies for `terminal` are selected by the OS (second) parameter in a build call to **mk9**.

o. I enhanced **mk9** to take a parameter indicating whether a Windows or a non-Windows OS build is wanted. Several other options have been changed, in name and effect (see the `HOWTO` file, 'Building Your Own Version of **ee9**').

p. I changed **ee9_self_test**, and its ancillary routines, to make them compatible with Windows/Cygwin.

q. I updated all the documentation to take account of Windows deployment, and I added a new Appendix to the **ee9** *Users' Guide*, giving various examples of the use of **ucc**, **nine**, **whet** and **tsd**.

r. I converted all the documentation that was formerly in simple text files into PDFs, as the easiest way to circumvent the—eccentric—handling of plain data in Windows.

V1.9e, 10-Dec-2011: Third public release

a. I advanced the version number from 1.5w to 1.9e, in view of the large number of improvements.

b. I added a feature, `AUTHENTIC_TIME_MODE`, whereby the real elapsed time is made to approximate the elapsed time that would have been taken by a real KDF9 (by inserting timed pauses at suitable points during the run).

c. I changed the `N` option designator to `A`, for '**a**uthenticity', and added `AUTHENTIC_TIME_MODE` to its options.

d. I reconceptualised 'verbosity' mode as 'visibility' mode, and added two new visibility flag options: `D` to request that optional **d**ebugging messages be output, and `T` to request that `AUTHENTIC_TIME_MODE` be set.

e. I fully implemented the TSD's device (de)allocation `OUT`s, using `CPDAR` properly to prevent access, in problem program mode, to unallocated devices.

f. I improved the way the `FW` package deals with prompts, with a resulting simplification of the two `OS_specifics` packages.

g. I amended the `FW` package to support authentic timing of typewriter output.

h. I provided implementations of the `P[IO][ABCD]` and `PM[ADF]` orders for the fixed disc drive, based on information from David Holdsworth, and on hypotheses developed by analysing `P198` in the Eldon 2 Director and further by extrapolating from the Data Products 5022 disc drive manual and the I.C.T. 1956 disc drive flyer.

i. I enhanced the `IO` package to support fixed disc emulation and authentic timing of slow device output.

j. I improved the traced operand value for instructions that update a `Qk`, as opposed to a `Qq`, operand.

k. I tidied up the procedures that deal with LIVs.

l. I rephrased the error message produced by a parity error, to make it clearer that it is probably only due to reading past the end of data.

m. I added support for 'character' (in reality, *one character per word*) transfers to the `TR`, `TP` and `FW` devices, so that `PIC`, `PID`, `POC` and `POD` are now fully implemented by the two-shift devices.

n. I corrected the timing of transfers on two-shift devices, allowing properly for shift characters, for filler characters, and for the fact that `FW0` has separate input and output streams.

o. I improved the realism with which short loops and short-loop jumps are timed, removing some ugliness in the process. The CPU time is now monotonically non-decreasing.

p. I implemented `POF` for the `TP`, on the basis of the only reasonable hypothesis I can make of its action.

q. I implemented all documented orders for the `CR` and `CP` devices.

r. I added a **c**ard-code dumping format, code `C`; a **l**ineprinter-code dumping format, code `L`; separate Case **N**ormal and Case **S**hift formats, codes `N` and `S`; and a Latin-1/**A**SCII dumping format, code `A`, intended to facilitate the use of 'character' transfers with Latin-1 data.

s. I somewhat tidied the `state_display` package, which 'just growed'. Ditto, the various `IOC` packages.

t. I changed the `Qp` terminology to `Qk`, for consistency with the Manual and KAL4.

u. I radically reworked the instruction-fetch logic in order to remove a kluge in the implementation of the `=EeQq` instruction. (It was needed to prevent Director from overwriting the executing instructions in a `JrCqNZS` loop that zeroizes the whole of core.) Instead, I implemented a much closer approximation to the logic and the instruction buffer registers (IWB0 and IWB1) of Main Control. Partly in consequence, the virtual CPU time for the Whetstone Benchmark now much more closely approximates the actual measured time.

v. I added a new **test program mode** of execution, which operates like problem program mode, but runs the code in Director state. This is not authentic, but is very useful for executing 'hardware' test programs. I added a **nine_test** command to the `Testing` directory, to make running in test program mode more convenient, and changed the **ee9_self_test** command to use **nine_test** in place of **nine** for the `MT` test programs.

w. To better exploit the new device allocation feature (point e, above), tape reader 0 is detached from the binary program file after the (initial or overlay) program load, and re-attached to the `TR0` text file. This makes it possible to run a program that has two separate paper tape input streams.

x. I corrected various bugs in the `SHAD` and divide orders, revealed by David Holdsworth when running KAL4. I took this opportunity to tidy up the code for shifts, multiplications, and divisions.

y. I now set `E7` of test-programs and problem-programs to a string, in the format `DD/MM/YY`, the date some multiple of 28 years ago; so that 1962 < 19`YY` < 2000, and 19`YY` has the same date/day-of-the-week mapping as this year.

z. I corrected `OUT` 9 to deliver the time-of-day (instead of the elapsed time), and implemented `OUT` 17 to return both the CPU time and the 'notional' elapsed time.

α. I corrected a bug in `INTQq`, which was limiting the hold-up to 1 virtual elapsed second in non-boot modes.

V2.0r, 30-Jun-2015: Beta release of V2

a. I advanced the version number from 1.9e to 2.0r, to indicate the first release of a major new mark that includes graph plotting; updated the copyright date to 2015; and updated the acceptable system name parameters of **`mk9`**.

b. I changed the way that `OUT`s are logged in the retrospective traces, to make `OUT 8` more conspicuous and its parameters more perspicuous. Small integers (OUT numbers < 64) are logged in decimal; large integers (parameters > 2^47) are logged in octal; and other values are logged in Q store format.

c. I amended the behaviour of `OUT 8` when addressing a Flexowriter, to conform more closely to the behaviour of the TSD as notified to me by David Holdsworth. Each message is now truncated to 8 words, if need be; embedded Line Shift and Horizontal Tab characters are disallowed; "**`;`**" characters must be in bits 42-47 and must not be in the last word of the transfer; but anything after an End Message can safely be ignored. Failures are logged as error "730", in the manner of the TSD. A correct message is always typed on a fresh line, a new line being taken if the typing position is not there already.

d. I corrected some minor portability problems in the **`mk9`** script and added support for the Raspberry Pi platform under Raspbian Linux.

e. I added a new **non-interactive mode**, set with the new option **`N`**, to adapt **`ee9`** to runs invoked by a command script. In this mode it is not possible to supply responses to prompts, whether from the KDF9 program or from **`ee9`** itself; so if an interactive input is requested in non-interactive mode, **`ee9`** terminates with a suitable diagnostic message. Non-interactive mode can also be set with the command line miscellany parameter.

f. I provided an implementation of the model 564 Calcomp graph plotter, as described in Appendix 6, §5, p.302 of the Manual. There was provision on the KDF9 to switch a buffer manually between a tape punch and a graph plotter; in **`ee9`** this is done with a new settings file option, **`G`**. If `G` is given, `GP0` replaces `TP1`. This can also be done with a miscellany parameter including **`g`** on the **`ee9`** command line.

The drawing is output to the `GP0` file in the form of Encapsulated PostScript. I expect any competent modern OS to have a utility that will open such a file and display the plot on screen.

g. It was possible to fit pens with a variety of ink colours and ball-point tip sizes to the plotter. To simulate this, option `G` can set the pen colour from the list: `Black` (the default), `Blue`, `Brown`, `Cyan`, `Dark_Blue`, `Dark_Cyan`, `Dark_Green`, `Dark_Grey`, `Dark_Magenta`, `Dark_Red`, `Green`, `Grey`, `Magenta`, `Red`, `White`, and `Yellow`; and, additionally, can set the pen tip size from the list: `Extra_Extra_Fine` (the default, one plotting step wide), `Extra_Fine`, `Fine`, `Medium`, `Medium_Broad`, `Broad`, `Extra_Broad`.

h. I implemented OUT 5 for `GP0`, taking the device type code to be #20, as stated in "Order Code Notes 18-Further OUTs" and consequently changed the output format of device type codes from decimal to octal.

i. The Usercode program `GPT.k3` in the Assembly directory runs a test on the plotter. The Usercode program `TR2GP.k3` in the Assembly directory copies a file of plotting commands from a tape reader to the plotter.

j. I added a new mode, **`D`**, to pseudo-Usercode core printing, such that operand and jump target word addresses are shown in **d**ecimal instead of octal, for ease of use with other software that works with addresses in this format.

k. I provided decimal interpretations of the internal registers, SJNS and Q store in the final state display, for use with the previous facility.

l. I added a decimal Q store interpretation of whole words in the NEST and core prints, and some whitespace to make for easier reading of their listings.

m. I modified the syntax of options with address parameters in the settings files to allow both octal and decimal addresses to be specified, octal addresses beginning with '#', as before.

n. I removed the largely ineffectual checking of the miscellany parameter from **`nine`**, **`nine_test`**, **`tsd`** and **`whet`**; the checking is now done by **`ee9`** itself.

o. I made a number of other minor improvements, e.g. to logging and error messages; and renamed the external tracing file as `trace.txt` instead of `full_trace.txt`.

p. I implemented the PHU store.

q. I modified the command line syntax for better mnemonic significance.

r. I refactored `FD` support packages to avoid a cyclical dependency.

s. I added a clearer interpretation of the parameter of an FD order in instruction traces.

t. I implemented CRLF line terminators for text-file output under MS Windows.

V2.1y, 28-Feb-2017: fourth public release

a. I advanced the version number from 2.0r to 2.1y, and the copyright date to 2017.

b. I enhanced the source to exploit features of Ada 2012, principally by converting `Assert` pragmas to pre- and post-condition aspects, and by using conditional expressions and case expressions wherever beneficial.

c. I corrected a minor programming error in **`kal3`** that evoked annoying warning messages from the C compiler.

d. I corrected the `FW` typed character count, which had been wrongly omitting the output of `LS` characters because of an inconsistency in the use of `put_EOL`.

e. I eliminated irrelevant registers from the information output during (pause mode) single-stepping, and changed the name used for the top cell of the SJNS from `S1` to `JB` (Jump Buffer), as it was called by EE.

f. I improved the *User Guide* by moving lengthy examples to Appendices and adding more explanatory context.

g. I enabled pre-run core printing for the Whetstone controller, overlaid on the Translator by OUT 1.

h. I corrected errors in both the documentation and the implementation of the `V` miscellany setting, and allowed for a command-line time limit.

i. I improved the handling of premature EOF/absent file errors when loading or booting.

j. I made the `IO` package a child of `KDF9`.

k. I improved KDF9 elapsed-time handling to prevent a possible overflow. This was extremely unlikely to happen on the actual hardware, which would have to had run for over 3 months without a break, but is much more likely in emulation, as 3 KDF9 months pass in as little as 9 hours real time!

l. I added support for time zones / daylight saving to the implementation of the time-of-day clock.

m. I corrected the machine codes for the `PMGQq` and `PMHQq` orders, following the discovery of a source text of the native Usercode compiler.

n. In response to the information that the Eldon KDF9s allowed the `CTQq` instruction in problem program state, I added this feature to execution in lax mode in anticipation of running Eldon some day; and I corrected the requirement for `MANUALQq` to be executed in Director state.

o. I added a **`Q`** command to option files. It causes **`ee9`** to **q**uit after loading the KDF9 program and before executing it. With a pre-run dump command in the same option file, this provides a convenient way to get a disassembly of the program without running it.

p. I added an **`X`** command to option files. It causes **`ee9`** to run in e**x**ternal trace mode, but giving the signature hash, [N1] where applicable, and the dis-assembled instruction.

q. I added **`q`** and **`x`** options to the miscellany parameter, with the same effect as the **`Q`** and **`X`** settings-file flags.

r. Following discoveries by David Holdsworth about the Kalgol compiler's use of the `FRB` instruction, its emulation has been modified to deliver a compatible result, despite that being at variance with restrictions stated in the Manual and in EE document K/GD.y.80, "KDF 9: SHIFTING AND SHIFT CONTROL".

s. I simplified and thereby corrected the tracing of an `OUT 8`.

t. I corrected the documentation with regard to the usage of `TR0` and `TR1`.

u. I extended the effect of the output suppression parameter to `OUT 1` overlaying and to the macros `tsd`, `whet`, `nine` and `nine_test`; including suppression of the output file listings with the `z` option.

v. Following a bug discovered by Bill Waite, I corrected the handling of the end-of-file condition on `TR` and `CR` input, when the transfer was not to End Message but the last word was incomplete due to inadequate data.

w. To make it as simple to run programs with `CR` input as with `TR` input, I added the command **`crnine`** (like **`nine`**) and a subcommand **`crud`** (like **`lud`**) and added a `CR`/`CP` test to **`ee9_self_test`**.

x. I removed all restrictions on overlaying (by means of `OUT 1`) programs other than the Whetstone Controller, and allowed for programs composed of multiple C-blocks.

y. Following problems experienced by Bill Waite, I changed the terminator for character-mode transfers to End Message, from a character with code 75_8 ("=" in Latin-1) to "|", which is the same as for fixed-length transfers and much more useful in practice.

z. Following problems experienced by Roberto Sancho Villa, I rewrote the `RLT` program to take a less fragile data format, and amended its data file accordingly. There were consequential changes in the `OUT4`, `OUTX` and `TRB` programs, and thence to the self-test. I also improved the somewhat misleading error message, particularly in Director mode, that was given when an attempt is made to operate an absent I/O device.

A. I restored the `FD` test programs `TFD`, `TFDEM` and `TFDFH` to working order and added `TFDOR` to test the area overrun (trying to read/write past the end of a seek area) logic.

B. I corrected a long-latent bug in the checking and reporting of an invalid I/O transfer store address.

C. I corrected the calculation of disc rotational latency, and added a Q store printing format for non-seek orders.

D. I changed the logic of LIVs so that test mode is treated the same way as program mode, avoiding misleading "Director failure" error messages from misbehaving test programs.

V2.1z, 08-Feb-2018: Ada 2012 corrected version of V2.1y

a. I advanced the version number from 2.1y to 2.1z, and the copyright date to 2018.

b. I corrected the error in JrV/JrNV that failed to clear the V bit.

V3.1a, 16-Jul-2018: Fifth public release

a. I advanced the version number from 2.1y to 3.1a; and made **`ee9`** announce its genesis in its opening message.

b. I significantly refactored the GP/plotter/postscript subsystem to enforce normal coupling between the packages; some former kinds of erroneous behaviour are no longer possible.

c. I continued the effort to use Ada 2012 to simplify the code, in particular exploiting the *name*`'Image` feature.

d. I improved the error handling in `host_IO`.

e. I shuffled the buffer numbers around so that the block devices are all in the 8..15 range; and included DR0 in the configuration for the first time, on buffer 15.

f. I removed a latent bug in the checking of buffer numbers for the TSD MT OUTs.

g. I corrected the I/O trace logging of TSD OUTs 6 and 7.

h. I completely rewrote the MT subsystem, removing all practical limits on the size of erased tape gaps, and increasing the limit on the largest block to more (4096 words) than the maximum recommended in the Manual (3000 words). Data blocks and erased tape are represented by one or more short 'slices' of data or erasure. This is more efficient for OUT 8 tapes than the previous implementation. The emulation of tape operation times is now much more realistic.

i. I implemented the DR0 device, making many assumptions as to its undocumented characteristics, but assisted by the annotated copy of a drum-capable Director from David Leigh. I added the test program TDRUM; and made the end-of-run statistics reporting for FD0 consistent with DR0.

j. I introduced new exceptions—`operator_error`, `operand_error`, `Director_failure` and `Director_operand_error`—to replace inauthentic LIV reports with more accurate error messages.

k. I added a new O option to enable the output of the current m/c state prior to an OUT 1 overlay.

l. I refactored IOC.FD along the same lines as IOC.DR, removing potential cyclical dependencies; and implemented a better stab at the working of the fixed-head disc orders, adding several new test programs TDISK*.

m. I made the elapsed time of the FD PMD order more realistic.

n. I made the buffer-status testing order PAR take its MC time; this ensures that it appears correctly positioned in the elapsed timeline.

o. I implemented the FD 'next sector' orders: PIE, PIF, PIG, PIH, POG, POH, POK, and POL. Their traces include the sector number actually used in the transfer.

p. TR0 and TP0 are no longer represented by the standard input and output; instead they bind to device files in the same way as other devices.

q. I made it possible to choose between Latin-1 and KDF9 code for two-shift devices, on a per-device basis, with new **ee9** parameters **-**TR and **-**TP; furthermore, a program file to be loaded or booted is now specified with a parameter of the form **+***program_name*, and not supplied as standard input. Paper tape input in KDF9 code is checked for validity, and a new exception—`invalid_paper_tape_file`—is raised when a problem is found.

r. I updated the ***nine*** and **whet**-related command files to support the new command-line syntax of **ee9**.

s. I improved program loading from paper tape, allowing loading up to the 32K limit, and checking the data for consistency and feasibility; the exception `invalid_paper_tape_file` is raised on premature EOF.

t. I changed the code for booting and loading to expect program files in authentic KDF9 paper tape code. I wrote a program (see the following point) that converts raw bytes to paper tape code, and incorporated it into the **ucc** command, so that the user need not see the raw byte files. For this reason I also removed the `.kdf9` filename extensions from programs generated by **ucc**, as well as those originally in the Binary directory. This makes the names of WAlgol, KAlgol, Director and OUT 1 overlay files consistent with program files in general, and is a step towards being able to load problem programs under Director.

u. For convenience in dealing with KDF9 code files I wrote two simple utility programs. These are:

a_block: output a program call tape, or 'A block', in KDF9 code; take the 'program reference number' and 'title' from successive lines of standard input, and write the result to standard output.

a2b: read from standard input in a stated code and write it to standard output in another code. Conversions are implemented from raw bytes to paper tape code; from paper tape code to octal words in half/word, Q store and syllable formats; and between paper tape code and Latin-1.

v. Following a more careful reading of §8.4 of the manual, assigning to Q0 is now **always** simply overridden, either by setting it back to 0, or by not changing it from its 0 value in the first place.

w. Following information from David Holdsworth, I corrected a bug in the `JrV` order, to make it clear `VR`.

x. I added the OUT 8 'gap' feature, previously missing.

y. I wrote a command, **tsdnine**, to simplify running problem programs under the Time Sharing Director. The **rlt** shell procedure is first run by **tsdnine** with its usual data files, to ensure that the `MT?` files are set up for Director use. **tsdnine** then takes its first parameter to be the name of a Usercode program in the `Assembly` directory, assembles it using **ucc**, creates an A block for it using **a_block**, concatenates this with the output from **ucc**, and any data file, and writes the result to `TR1`. It then invokes Director using the **tsd** command. Once Director has booted, a `FLEX` interrupt (^C) makes it load and run the program from `TR1` (the Director load command has been added to `FW0_for_Director`).

z. The retro-trace of I/O operations now logs the instruction in the device-specific form given in the Manual, if such is defined; e.g. `PIFQ1` for a magnetic tape is listed as `MBREQ1`.

A. I refactored the package `OS_specifics` so that it no longer depends on any other **ee9** package. This results in a cleaner interface, and allows it to be used by the **a_block** and **a2b** programs without dragging in most of **ee9** via unwanted transitive dependencies. There is a trivial consequent change to `ioc-two_shift-fw.adb`.

B. I corrected an error in `ioc-slow.shift.get_symbols` that generated unnecessary case shifting for characters that are available in both shift- and normal-case character sets.

C. I implemented a means to 'poke' amendments into binary programs after loading and before the start of execution. This required the 'postmortem' `P` setting to be renamed `F`, and the removal of the (almost never used) fetchpoint-only setting. Fetchpoints can still be set in combination with storepoints, using the watchpoint setting.

D. I updated the historical paper '*The Hardware of the KDF9*'.

E. I put in some simple infrastructure for debugging runs: the tracing option **d** sets the Boolean variable `debugging_is_enabled`, which can be tested at run time to make output conditional. Enclosing a call of a procedure in **pragma** `Debug(…)` ensures that its code is included in the build of **ee9** only by a 'warning' compilation; in a normal compilation it is completely suppressed and incurs no space or time overhead. The new `tracing.message` procedure displays its string parameter iff `debugging_is_enabled`, and its body is included in the object program subject to **pragma** `Debug(…)`.

F. I refactored the `IOC` package and its descendants, and revised the I/O device type hierarchy to be more reflective of the similarities between the actual devices, thus:

```
IOC.device (abstract—objects cannot be declared)
  IOC.fast.device (abstract—objects cannot be declared)
      IOC.fast.DR.device
      IOC.fast.FD.device
      IOC.fast.magtape.device (abstract—objects cannot be declared)
           IOC.fast.magtape.MT.deck
           IOC.fast.magtape.ST.deck
  IOC.slow.device (abstract—objects cannot be declared)
      IOC.slow.shift.device (abstract—objects cannot be declared)
          IOC.slow.shift.FW.device
          IOC.slow.shift.GP.device
          IOC.slow.shift.TP.device
          IOC.slow.shift.TR.device
      IOC.slow.unit.device (abstract—objects cannot be declared)
           IOC.slow.unit.CP.device
           IOC.slow.unit.CR.device
           IOC.slow.unit.LP.device
```

G. I factored the KDF9 symbol type and character codes out of `KDF9` into their own package, `KDF9_char_sets`, so that they could be accessed by **`a_block`** and **`a2b`** without dragging in the rest of `KDF9`.

H. I removed all `Unsuppress` pragmas from the source files and put them into the compilation configuration files. This means that packages formerly compiled with suppressed checks now compile with them enabled. This has made no perceptible difference to the performance of **`ee9`**.

I. I corrected a long-standing error in the `KDF9.CPU.reconstruct` operation, which was yielding an incorrect double-length floating-point result when given a fixed-point fraction with an all-zero more-significant word.

J. I changed the interpretation of the graph plotter's command codes, in accordance with evidence found in manuals for contemporary computers that employed the same device.

K. I amended the DIVR and DIVD microcode to correct an error in calculating quotients (and hence remainders) due to assuming that the Ada fractional division operation truncates, when in fact it rounds; and improved the handling of overflows and of invalid dividends having D0 of the less significant word non-zero. The latter is almost certainly not authentic, but is better than nothing.

L. I removed some unused presentational floating-point functions from `KDF9.CPU`.

M. I allowed more OUT 8 stream numbers, and shared the #10-17 range between TP0 and TP1.

N. I updated the Guide and added Appendices describing the TSD OUTs. I detailed therein the ways in which the authentic OUT 8 operations differ from the approximations made by **`ee9`**.

O. Since the **`ucc`** command now produces object programs in KDF9 character code, which is not acceptable to David Holdsworth's **`kdf9`** emulator, I provide the command **`uc2kdf9`** to produce a program in the form needed by **`kdf9`**.

P. I corrected an error in the way some invalid operations were treated in test program mode and in Director state. They were ignored, which could cause a later failure in a routine that relied on invalid operations forcing termination to ensure a precondition. Now all invalid operations force termination.

Q. I added a DIVR test to the self-test.

R. I added Appendix 8 to the *Users' Guide*, listing and categorising all terminating error conditions.

V3.2K, AUGUST 2019: SIXTH PUBLIC RELEASE

a. I advanced the version number from 3.1a to 3.2**k**, to celebrate **ee9**'s first support for **K**idsgrove Algol .

b. I enhanced the **`a2b`** utility, to simplify the options flags; changed the `-p2o` flag to output a line/word number and the character interpretation of each word; and added a paper tape code to raw bytes conversion with flag `-p2r`.

c. I modified the **`a_block`** utility, to generate a call tape for a program on a specified medium rather than only a TR, and renamed it **`call_tape`**, which is more historically accurate.

d. I moved the character-mapping function `glyph_for` from `state_display` to `KDF9_char_sets`, for better modularity and for reuse (e.g. in **`a2b`**).

e. I changed the string-length field in a MT slice from 2 bytes to 1 bytes, thus avoiding endian-ness issues and making MT files more portable.

f. I added Appendix 9 to the *Users Guide*, listing the TSD TINTs and other console operating information, and updated the error messages in Appendix 8.

g. I implemented an instruction-word execution frequency profile with histogram output, and added the **`H`** option to control it, and the previously existing instruction type histogram.

h. When disassembling, if the operand of a SET instruction needs more than 1 digit of the selected base, its octal representation is supplemented by its decimal representation in a comment; or *vice versa*.

i. I added a LIV when the EXIT instruction has an effective word address > 8190, in accordance with the EE engineering document K/GD.y.82, "INTERRUPTION AND FETCH NEXT INSTRUCTION SEQUENCES OF KDF9 MAIN CONTROL", §4.0(b)(iii).

j. I reverted the change allowing CTQ*q* in problem program state with lax mode, following new information from David Holdsworth about how the Eldon 2 Director actually worked.

k. I wrote the historical paper '*Software of the KDF9'*.

l. I improved the treatment of the clock, and clock interrupts, when running in boot mode. Directors should now report *somewhat* more accurate CPU and elapsed times.

m. I tidied up and pruned confirmatory chatter in the log about OUTs.

n. I corrected an error found by David Holdsworth, whereby backwards reading (MBRQq) of magnetic tape blocks gave incorrect results for the first data block following the label.

o. I corrected a second error, when the data block on tape was shorter than the core transfer area defined by `Qq`. As a consequence, the `reflect` procedure, formerly in `KDF9.store`, is no longer needed.

p. I corrected a third error, whereby the MWEQq order did not write the transfer-terminating EM to the MT file.

q. I added Appendix 10 to the *Users Guide*, describing the data structure used in the implementation of MT files.

r. I wrote a suite of small Usercode programs `OUT4*`, for a more comprehensive test of the MT subsystem. These programs can be conveniently run as a suite with the **`run_mt_tests`** shell command. Some of the tests are designed to run to completion; others are intended to check that error conditions are detected and handled correctly. Proper execution of each test in the suite is indicated by the output to TP0 of a final line with the word `FINISH`.

s. I changed the maximum MT reel length from 2400 feet to 3600, as used by Atlas 1, for lack of better information.

t. I corrected an error that gave the wrong CPU time usage for MT I/O transfers.

u. I simplified the implementation of the drum device, DR0, by getting rid of the in-RAM cache and doing all drum I/O direct to the DR0 file.

v. I wrote a suite of small Usercode programs `TDRUM*`. These programs can be conveniently run as a suite with the **`run_dr_tests`** shell command.

w. I did the same thing for the FD0 tests `TDISK*`, and provided a shell command **`run_fd_tests`**.

x. I wrote a utility program, **`mtp`**, to provide a legible MT file analysis. For details, see Appendix 10 of the *Users Guide*.

y. I provided a shell command, **`kids`**, to run the Holdsworth-Huxtable-Wichmann recreation of the Kidsgrove Algol compiler and runtime environment. For details, see 'GROOVING WITH KIDSGROVE' in the *HOWTO* file.

x. I wrote a suite of small Usercode programs **`CRT*`** and **`CPT*`** to test card reader and card punch emulation. These programs can be conveniently run as a suite with the **`run_cr_tests`** shell command.

y. I made a number of small corrections to the *Users Guide*, in particular to Appendix 2B.

z. With a view to better support for running KAlgol object programs, I made the new directory `Testing/Data` the normal place to keep data files (instead of `Testing/Assembly`).

A. I corrected an error in the display of very large floating point operands in the retrospective trace,

B. I added a decimal interpretation of Q store values to the single-stepping witness print.

C. I corrected an error in processing a watchpoint request in settings files when only a starting address was given.

D. I wrote the historical paper '*Benchmarking and the KDF9*'.

E. I changed the maximum MT slice size from 125 to 128 bytes, to represent KAlgol work tapes more efficiently.

F. I finalized the document '*A KDF9 Bibliography*'.

G. I corrected an error in addressing a halfword with a negative index Mk in MqMkH-type orders.

H. I decoupled the setting of hardware allocation bits in CPDAR from the availability of devices to OUT 5, this being necessitated by the resurrected KAlgol, which writes to LP0 both directly and via OUT 8 spooling.

I. I provided a shell command, **`k9`**, that runs both the KAlgol compiler and its object program with **`kdf9`**.

J. I added a miscellany option **`W`** to suppress FW0 output; this is useful for running KAlgol under **ee9**.

K. I provided a comment facility in settings files: any line starting with — or `/` is ignored.

L. I added a 'here document' facility to **ee9**, to provide TR0/1 and CR0 input when an input file is exhausted or absent.

M. The all-zero syllable was formerly treated as in invalid order, but evidence has come to light that the hardware actually treated it as a "no-op", like DUMMY, so **ee9** now does the same, identifying it in traces as DUMMY0.

N. I corrected an error in floating point multiplication whereby -0.5 squared gave +1.0 instead of +0.25.

O. I enhanced the feature whereby a file can be nominated to supply TR0/TR1 or CR0 with input when reading encounters the end of data, either because no input file was named, or because its contents are too short. Rather than taking a single letter, it now allows a name *n* to be given, nominating the file `Data/`*n*`.txt`.

P. I changed the handling of preposterous scale factors in FLOAT and FLOATD orders. They now use only the least significant 8 bits of the N1 operand. This is undocumented behaviour, revealed by testing the `sqrt` function of Kidsgrove Algol, which deals with the scale factor differently from the Whetstone function.

Q. I tidied up **`nine`**/**`crnine`**/**`whet`**, and their auxiliary commands, improving error messages and removing code that has accumulated since time immemorial for diagnostic and other purposes, but is no longer needed.

R. I added the fixed-point interpretation of nest cells to the witness print provided during single-stepping.

S. I provided a new function to compute the time required for implicit shifts made during normalization, which are slightly slower than explicit shift operations.

T. I implemented the non-escaping underline on FW output, using ANSI terminal SGR escape codes.

U. I allowed for newer **M**icrosoft terminal programs by making the output of ANSI terminal SGR escape codes—for red and for underlined Flexowriter output—a dynamically-enabled option, disabled using the miscellany/visibility flags `m` or `M`, the default setting now being to **enable** output styling, even on Windows. I also moved the escape codes from `OS_specifics.ads` to `IOC.slow.shift.FW.adb`, where they now fit better.

V. I updated *ee9 Implementation Overview* to describe divisions, shifts, and Flexowriter output improvements.

W. I added a test of underlined FW output, and a test of Kidsgrove Algol, to **`ee9_self_test`**.

X. I added three bogus OUTs. OUT 99 returns the value of the Instruction Count Register in N1, to help instrument the execution of KDF9 machine code programs. OUT 98 switches off authentically-red printing by the Flexowriter if N1 contains 0, to provide a convenient method of transcribing original-format Algol programs for publication. OUT 70 is used for diagnostics in KAlgol, and is treated as a no-op by **ee9**. These "OUTs" are not documented in the *Users Guide* and may be withdrawn at any time!

Y. I enhanced the procedure `load_a_program`, in the package `IOC.slow.shift.TR`, to accept and check A blocks—which remain optional—at the start of object programs.

Z. I enhanced **`a2b`**, to provide a way of extracting clean text from KAlgol TP0 Usercode output.

V4.1D, 31-JAN-2020: SEVENTH PUBLIC RELEASE

a. I advanced the version number from V3.2k to V4.1**d** to celebrate much better support for running **D**irector.
b. I improved TSD's timing by setting the (inauthentic?) `CLOCK_interrupt` bit in RFIR when effecting a K4 order.
c. I removed a potential LIV from `has_locks_in`.
d. I prevented a potential LIV when doing a store print by ensuring that NIA is not set to 8191 during flow analysis.
e. I enhanced the **`mtp`** utility to give a fuller and more accurate interpretation of OUT 8 data.
f. I renamed the `lax_mode` option to `faithful_mode` and made it the default nest checking option. The former default, `strict_mode`, is now intended to be used only for debugging i*n extremis*.
g. I improved the way the test (METQq, *etc*) operations were implemented *vis-a-vis* lockouts, which was causing problems in boot mode, specifically not returning to the problem program after OUT 8.
h. I enhanced the **`mtp`** utility to 'print down' OUT 8 streams.
i. I truncated the times returned by OUTs 3, 9 and 17 to a multiple of 32μs, the clock 'tick' of the hardware.
j. I corrected the error message, in the event of RESET caused by jumping to a syllable number of 6 or 7.
k. I enhanced the **`a2b`** utility to generate call tapes from object program binaries.
l. I moved several magnetic tape and OUT 8 declarations to the package `magtape_shared_declarations`, making them available to both **ee9** and **`mtp`**.
m. I wrote a Usercode program, **`PLT`**, to create and update program library magnetic tape files.
n. I enhanced the **`tsdnine`** command to work with PLTs.
o. I fixed a bug in MFSKQq, which was not stopping at an LBM-marked block.
p. I removed an instruction-fetching bug that caused an infinite loop in Director when an interrupt occurred immediately after executing a successful JrCqNZS instruction.
q. I moved previously scattered information about ancillary programs to the new Appendix 11 of the *Users Guide*.
r. I increased the maximum magnetic tape block size to 32768 words. A larger block size being logically impossible, this cannot be exceeded by any program.
s. I fixed an error that made TSD fail in authentic timing mode.
t. I made Flexowriter output look more realistic in authentic timing mode.
u. I corrected a latent error in the calculation of the 'next' sector in `ioc-fast-fd.adb`.
v. I improved the diagnostics on program termination, particularly when the `A` or `E` visibility option is in force.
w. I removed a "dont't care" condition from the decoding of the SET order, so that all invalid encodings now LIV.
x. I made invalid transfer addresses in I/O orders LIV, rather than vaguely failing due to an "invalid operand".
y. I prevented DUMMY0 orders from disrupting Usercode-format core prints.
z. I changed the compiler used to translate **`kdf9`** and **`kal3`** for macOS (only) from **`gcc`** to **`cc`**, Apple's LLVM compiler. This avoids library-positioning conflicts with Xcode.
A. I improved the disassembly of jump and fetch/store orders by giving both octal and decimal word addresses.
B. I wrote a set of Usercode programs (**`LIV?`** and **`NOUV?`**) and shell commands (**`run_liv_tests`** and **`run_nouv_tests`**) to run a suite of basic CPU functionality tests, and added them, along with the extant I/O functionality test suites, to **`ee9_self_test`**.

V4.2F, 21-JUN-2020: EIGHTH PUBLIC RELEASE

a. I advanced the version number from V4.1d to V4.2**f** to reflect the victory of **f**aithful mode.

b. I removed all trace of the strict mode of operand checking, including the `l` miscellany parameter; **ee9** now always runs in the former faithful mode, and, if not with authentic timing, then in `modern_times_mode`.

c. I made `read_KDF9_tape_code` not fail at premature EOF; it now only sets the buffer abnormal.

d. I factored the library-level procedure `enable_GP0_if_requested` out of `ee9` and `KDF9.Directors`.

e. I renamed `magtape_shared_declarations` as `magtape_data` for brevity.

f. I added procedures `trap_unimplemented_feature` and `trap_invalid_paper_tape` to `KDF9`, to make error handling more consistent and modular.

g. I made Usercode-format core prints work well for Directors as well as problem programs.

h. I rejigged the method by which ^C is handled, using a atomic Boolean variable that is tested in the instruction interpretation loop, to avoid a race condition with native KDF9 interrupts.

i. I corrected the return address saved by the INTQq order from CIA to NIA, and improved the non-boot mode semantics to more closely reflect the behaviour of the TSD.

j. I changed lockout setting and clearing to better imitate the behaviour of I/O Control, thus correcting an error that sometimes left spurious lockouts enabled when running TSD.

k. I improved the behaviour of backward skips with an empty MT, so that an error is reported and not an emulation failure.

l. I improved the tracing of I/O events and of interrupts, giving more informative messages and including their TSD context.

m. I corrected spurious LIVs in boot mode, when handling lockouts for I/O operations in very small programs.

n. I renamed **`ee9_self_test`** to **`ee9_reg_test`** in recognition of the fact that it has long been a suite of regression tests.

o. I suppressed flow analysis unless a Usercode-format core dump is wanted, and corrected core dump addressing in boot mode.

p. I implemented 'IBM-compatible' Ampex TM-4 7-track tapes, in the process simplifying the structure of the MT ADT.

q. I enhanced the **`mtp`** program to work with 7-track tapes and improved its output, in particular for OUT8 printer streams.

r. I wrote a placeholder implementation of the Standard Interface buffer (SI0), pending fuller information about it, but based on minimal reasonable assumptions as to its functionality.

s. I provided a way to choose between FD0 and DR0, and between ST0 and SI0, at run time, with a settings file option.

t. I corrected a possible overflow in `IOC.slow.shift.TR.read_KDF9_tape_code`, when trying to read into E32767.

u. I added the SI, the PDP-8 buffer, and the Ferranti tape punch, to the list of known device codes in TSD API emulation.

v. I restored the maximum MT reel length to 2400 feet from 3600, having found a definitive reference.

w. I changed POFQq, 'word gap', for the TP and SI devices, to produce 8 times as many leader frames as POEQq.

x. I wrote a small set of Usercode programs and a shell command (**`run_si_tests`**) to run a suite of basic functionality tests for the interim SI device implementation.

y. I enhanced `host_IO.reattach` to allow an input **and** output device such as SI0 to preserve its read/write access mode.

z. I made the creation of devices dynamic, by explicitly calling the type's `enable` procedure, to allow for configuring optional devices (and incidentally improving the diagnostic message when a device file is absent).

A. I reduced the maximum string length in a magnetic tape slice to 93 characters, resulting in more space-efficient representation of card images and OUT 8 line printer blocks, with little effect on the space efficiency of POST-sized blocks.

B. I corrected a bug that failed to lock out busy devices for some problem program transfers in boot mode.

C. I changed the codes used to represent the KDF9 End Message and Semicolon characters to be their 8-bit paper tape codes in 'character' transfer orders for the FW, TR, TP and SI devices. This is more authentic than before, if less handy.

D. I updated Kidsgrove Algol to the 2020/06/08 version at `settle.ddns.net/KDF9/kalgol/DavidHo`. It includes much-improved I/O libraries by comparison with the version in **ee9** V4.1d.

E. I improved the retrospective tracing of LOV interrupts with a message that more clearly indicates the reason for lockout.

F. I wrote a shell command, **`run_kids_tests`**, to automate regression testing of the Kidsgrove Algol compiler.

V5.1a, 04-Nov-2020: Ninth public release

a. I advanced the version number from V4.2f to V5.1a. Public releases of **ee9** are now 10 years old!

b. I removed a redundant copy of `say_a_polite_goodbye` from `execute`; it now exists only as a library-level unit and has been renamed as `say_goodbye`.

c. I did away with the HOWTO file as the relevant information is now in either the README file or the *Users' Guide*.

d. I added a call of **`neat`** to the **`kalgol`** command, to make the Usercode more readable.

e. I corrected wrap-around errors when dumping core to word 0.

f. I improved the diagnostics given for an invalid P option in a settings file.

g. I wrote **`kidopt`**, to set up the option bits for the Kidsgrove compiler in a convenient and understandable manner.

h. I enhanced **`kalgol`** to use **`kidopt`**.

i. I changed `POSIX.data_prompt` and `IOC.slow.reattach` to refrain from forcing the use of `Testing/Data`.

j. I reorganized the directory structure for Kidsgrove Algol, so that the source code is directly within `Testing/Kidsgrove` and the ancillary material is within `Adjuncts`; amended **`mk9`** and the shell files in `Testing` accordingly.

k. I improved the handling of errors when reading numbers in `settings.IO` and in `settings.save_poke_value`.

l. Following (j) I renamed `Testing/Algol` as `Testing/Whetstone`, putting the two Algol systems on an equal footing.

m. I overhauled **ee9** and the shell commands so that they look for their various kinds of file in locations given by environment variables, with default to the fixed pathnames established by changes (j) and (l).

n. I added extensive documentation about the Whetstone and Kidsgrove Algol systems, and their usage with **ee9**, to the *Users' Guide* and the *Implementation Overview*.

o. I amended the behaviour of output to LP0, making it compatible with the expectations of the Kidsgrove compiler in matters that are not well defined in the Manual, such as incomplete transfers and transfers containing multiple LS and PC characters.

p. A PC character is now treated as printing the rest of a whole page, pages being taken to be 66 lines long—a standard stationery size. For example, sending PC to terminate the third line of a page is treated as printing 63 lines, bringing the printing position to the top of a new page.

q. I removed Horizontal Tab from the line printer character set: it was used only by offline printing software.

r. I changed the transcription to character form of NEST cells in stepping output; it now uses the punched card transliteration, which provides a glyph for every 6-bit character, unlike the former line printer transcription.

s. I prevented an emulation failure when the response to a data filename prompt is empty.

t. I added the '`@`' prompt to enhance interactive data filename specification.

u. I now display execution times to the nearest millisecond or nearest second, as well as exact microseconds, in the final state.

v. I corrected the stated character code for 'last slice of block' in the *Implementation Overview* from `BEL` to `BS`.

w. I rewrote **`mtp`**, simplifying the logic, correcting an intermittent error when despooling, and tidying the despooled output.

x. I corrected an error in **ee9** whereby an initial erasure slice could be left when writing a block into an erased section of tape.

y. I simplified the internal data structure that represents slices of a magnetic tape block and made reading and writing tape marks conform exactly to the Manual's description.

z. I corrected the setting of CPDAR by the =K2 order, which had been only accidentally allowing programs to run under TSD.

A. I removed a possible range error from the calculation of a magnetic tape's position.

B. I improved the format of CPDAR's current state and made interrupt tracing messages clearer.

C. I tidied up some old code to my 2020 standards.

D. I changed the maximum magnetic tape slice size to 81 characters, which is optimal for card images and print lines.

E. I made the messages showing the final status of I/O devices appear in the `KDF9_log.txt` file; previously they were listed only to the terminal. `POSIX.output*` is now used only when interaction is involved.

F. I added facilities to **`a2b`** and **`mtp`** to interpret output in Ferranti 5-hole paper tape code.

G. I enhanced the `K` flag to allow any feasible I/O equipment configuration.

H. I corrected a failure to load a program that starts with an A-block and now allow an A-block to be absent, minimal, short or maximal, despite what is said in §26.3.1 of the KDF9 Programming Manual.

I. I changed the way **ee9** searches for a labelled tape, making it ignore empty tape files instead of failing them.

J. I changed the semantics of PMBQq (MBTQq) so that it does not set TR when the tape file is empty. This allows TSD to execute successfully with an empty tape file. (Without this change TSD got into a loop testing and retesting the deck.)

K. I fixed an error that made TP1 undefined if graph plotting was (pointlessly) specified on the **`whet`** command line.

L. I consolidated **`whet`**, **`nine`** and **`crnine`** by in-lining **`lap`**, **`dow`**, **`lud`** and **`crud`**, which have now been deleted.

M. I introduced a new 'absent' device type, with code `NA` for '**n**ot **a**ttached', to better handle any buffer without an assigned I/O device. This now causes an execution error if and only if an attempt is made to use such a buffer.

N. I now support the configuration of up to 8 magnetic tape devices, up to 2 devices of each of the CP, CR, LP, SI, ST, TP and TR types, and at most 1 device of each of the DR, FD, FW, and GP types. Including GP0 automatically switches the TP1 buffer to GP0 in the program modes, but the `g` option in the miscellany parameter or in `settings_1.txt` (or both) is still needed when running programs under Director, as the latter has no access to **ee9**'s internal state.

O. TP and SI output from the POEQq and POFQq "gap" instructions used to be completely suppressed. It is still suppressed for TPs in Latin-1 mode, but otherwise outputs a NUL character for each frame of runout that the hardware generated.

P. The **`a2b`** program now ignores NUL input characters when given the `-p2l` and `-p2t` parameters.

Q. I wrote a shell command, **`run_whet_tests`**, to automate regression testing of the Whetstone Algol compiler.

R. I enhanced the continuation file facility so that it now also works when a device is in KDF9 code mode.

S. I improved the tracing (in non-boot modes) of the I/O operations that effect OUTs 6, 7 and 8, and showed magnetic tapes rewinding after OUTs 0 and 2.

T. b. I corrected a finalization failure when a tape is mounted read-only.

U. When an unrecognised code is output to GP0 I now move the plotting position to the origin, instead of failing the run.

V5.2b, 01-Jan-2020: Tenth public release

a. I advanced the version number from V5.1a to V5.2**b**: it is the **B**rexit release and a celebration of the now inevitable **b**reakup of the United Kingdom, as **b**oosted by the "world-**b**eating" **b**ullshit of the **b**lithering **b**uffoon **B**oris (**b**oo!).

b. I corrected a finalization failure when a tape is mounted read-only, and retrofitted it to V5.1a, as it was so trivial to do.

c. I thoroughly overhauled the ways in which failures are treated and (non-KDF9-interrupt-caused) exceptions are raised.

1. Impossible I/O operations, that are requested by valid I/O orders applied to correctly addressed devices, are now handled by `IOC.trap_failing_IO_operation`, not the general routine `KDF9.trap_illegal_instruction`. In Director state `trap_failing_IO_operation` ends the run, because a serious malfunction is indicated. When running a problem program in boot mode it sets the device abnormal and abandons the order; the onus is on the program to detect this and act accordingly—if it does not do so it may LIV. In non-boot modes it ends the run with a specific diagnostic. These arrangements provide the maximum of diagnostic information without forcing a crash that would not have happened when running a program under Director in real life.
2. Benign failures that can be recovered from, with diagnostic output in place of an expected output, now do so recover.
3. All failures that might indicate a malfunction by **ee9** itself raise `emulation_failure` with a diagnostic.

d. I revised the build procedures to make them less verbose and to add a facility that better tidies up non-macOS distributions. I also created a shell command **`de40ine`** to help macOS users get their downloads out of the quarantine imposed by Apple's System Integrity Protection/Gatekeeper security measures.

e. I amended `IOC.fast.DR` to reflect discoveries made in SRLM, §103, Appendix 2, p.10-59-0, about the physical characteristics of the drum unit.

f. I amended `KDF9.Directors` so that **ee9** behaves in the same way as a **non**-TSD when an OUT 8 stream is sent to FW0, specifically: by decorating output messages with the '`[m] `' prefix, and query prompts with '`[q] `'.

g. I fixed an error in **`mtp`** that made some stream headers get printed more than once if the OUT 8 MT contained interlarded output for several different streams.

h. I changed the build process to use **`gzip`** for Linux, as the **`zip`** command seems to be absent from Debian.

i. I fixed a bug in `KDF9.Directors` that caused OUT 2 to fail, because it was clearing CPDAR before attempting to rewind the deallocated MTs. I also refactored the implementation of OUT 2 between `KDF9.Directors` and `KDF9.microcode`, to better trace the effect of OUT 2, particularly when pre-overlay diagnostics are requested. This required the introduction of a new exception, `program_restart`.

j. I added '`.`' to the miscellany parameter, for a time limit of 1E6 instruction executions, to speed up KAlgol regression testing.

k. I corrected the possible mis-identification of a non-DMAing buffer as being responsible for a main store lockout.

l. I tidied up the device-emulation test commands and renamed **`nine_test`** as **`nine_priv`** to avoid confusion with them.

m. I implemented OUT 16 in the manner of the non-Time Sharing Director, with a prefix of '`N⍰ `'.

n. I implemented the "Drum Director" OUTs: 11, 12, 13 and 14.

o. I added `k` / `K` to the miscellany parameter, making it possible to enable DR0 on the command line.

p. I added `b` / `B` to the miscellany parameter, making it possible to enable SI0 on the command line.

q. I corrected a SHAD right shift when the operand is invalid, in that D0 of N2 is 1, making it consistent with the hardware.

r. I implemented the fixed disc OUTs 41, 42, 43, 44 and 45.

s. I corrected the timing of seeks when more than one fixed disc spindle is used, and also when the fixed-head platter is used.

t. I allowed for up to 9 MT decks to be included in a configuration, following information from EE documentation.

u. I allowed the final "`.txt`" of a data file name to be omitted when it is supplied interactively, both for convenience and for consistency with data file names supplied on the command line to `nine`, `kids`, etc.

v. I now recognise '`q`', '`Q`' or end-of-file (e.g. by typing `^D`) given in response to a data prompt, with the effect of signalling an impossible I/O operation; see point c.1, above, for the possible consequences.

w. I moved support for TSD's DR, FD and MT OUTs from `KDF9.Directors` to the new packages `IOC.fast.DR.OUTs`, `IOC.fast.FD.OUTs` and `IOC.fast.MT.OUTs`. Support for OUTs 8 and 16 was moved out to the new child procedure `KDF9.Directors.do_TSD_IO`. These changes improve the cohesion of `KDF9.Directors` and make specific Director funtionality easier to locate.

x. I moved the functions `as_DR_command` and `as_FD_command` from `formatting` to `IOC.fast.DR` and `IOC.fast.FD`.

y. I moved the procedure `log_API_message` from `KDF9.Directors` to `HCI`, making it generally available.

z. I moved the procedure `fail_OUT` from `KDF9.Directors` to `KDF9`.

TO DO: complete the PLT program, verify multiprogramming with TSD, include newer KAlgol system, …