

## ee9 V8 ERRATA

- The POC order for the FW would output ‘<’ as ‘;’ when in ‘transcribing’ (i.e. 8-bit transparent) mode.

SIGNIFICANCE: MINOR.

- The disassembly of an operand as a symbol would fail when its value was outside the valid range of addresses.

SIGNIFICANCE: MINOR, AS IT DOES NOT AFFECT THE EXECUTION OF CORRECT PROGRAMS.

### V8.1A CORRECTS THESE ERRORS IN V8.0K.

- The disassembled operand of SET was shown in octal if decimal output was chosen and its value was 8 or 9.

SIGNIFICANCE: TRIVIAL.

- An unlikely case in single-precision floating-point division caused an overflow (Constraint\_Error) failure.

SIGNIFICANCE: MAJOR IF THAT CASE AROSE, BUT IT WAS VERY UNLIKELY.

- The Unix commands `tr` and `sed` on ARM Macs fail on Latin-1 characters outside the ASCII subset. Their usage in `extract_symbols`, `sttl`, `peep` and `neat` has been replaced by new Ada programs.

SIGNIFICANCE: MAJOR, BUT ONLY ON ARM-BASED COMPUTERS.

### V8.1S CORRECTS THESE ERRORS IN V8.1A.

- The TSD OUT 47 may cause a NEST underflow.

SIGNIFICANCE: MINOR, OUT 47 IS NOT USED BY ANY ORIGINAL OR RESTORED SOFTWARE.

### V8.1U CORRECTS THIS ERROR IN V8.1S.

- `glance` fails on Usercode of the form: `*J....`

SIGNIFICANCE: MINOR, NONE OF THE CODE IT WAS CREATED TO PROCESS HAS THAT FORM.

### V8.1V CORRECTS THIS ERROR IN V8.1U.

- `glance` fails (again!) on Usercode of the form: `*J....`

SIGNIFICANCE: MINOR, NONE OF THE CODE IT WAS CREATED TO PROCESS HAS THAT FORM.

- `bangs` and `kwote` call `ports` wrongly.

SIGNIFICANCE: MINOR, EASILY AVOIDED.

### V8.1X CORRECTS THESE ERRORS IN V8.1W.

- The presentation of NEST cells as strings is corrected, in the case that initial or final characters are blanks.

SIGNIFICANCE: TRIVIAL, THE CORRECT VALUE IS SHOWN IN OCTAL.

- A range error resulting from grossly excessive output to a FW has been prevented.

SIGNIFICANCE: MINOR, AS IT DOES NOT AFFECT THE EXECUTION OF CORRECT PROGRAMS.

- A spurious range error in virtual address calculation has been prevented.

SIGNIFICANCE: MINOR, AS IT DOES NOT AFFECT ANY EXISTING PROGRAM.

### V8.2V CORRECTS THESE ERRORS IN V8.1X.

- My mistaken attribution of the Easter Sunday calculation to Zeller, by dint of file names, has been corrected to give the credit to T.H. O’Beirne, to whom it is rightfully due.

SIGNIFICANCE: HISTORICAL ONLY, THERE IS NO CONNECTION WITH EMULATION.

- A zero-divide emulation failure when the divisor is the smallest representable floating point number has been converted to a KDF9 overflow indication.

SIGNIFICANCE: MAJOR IF THAT CASE AROSE, BUT IT WAS VERY UNLIKELY.

### V8.2Z CORRECTS THESE ERRORS IN V8.2V.

- There was an error in ee9 for 12 years whereby an arithmetic right shift of -1 by 47 or 48 places would yield -1 instead of 0. Fascinatingly, it went unnoticed until today!

SIGNIFICANCE: MAJOR IF THAT CASE AROSE, BUT IT WAS VERY UNLIKELY.

- I further improved the handling of division by very small unnormalized numbers, so that the special treatment introduced in V8.2z is no longer needed.

SIGNIFICANCE: MINOR, AS IT AFFECTS ONLY PROGRAMS THAT USE INVALID DATA.

- I prevented a failure in ee9 caused by a ridiculous shift length in double-length shifts.

SIGNIFICANCE: MINOR, AS IT AFFECTS ONLY PROGRAMS THAT USE INVALID DATA.

### V9.0P CORRECTS THESE ERRORS IN V8.2Z.

**THE FOLLOWING ARE PROBLEMS WITH THE KIDSGROVE ALGOL SYSTEM,  
NOT WITH ee9's EMULATION OF THE KDF9 ARCHITECTURE.  
THEY OCCUR ONLY IN OPTIMISED OBJECT PROGRAMS.**

- The Kidsgrove system gives wrong results for some programs that contain calls on library routines.

**SIGNIFICANCE: MAJOR, PREVENTS CORRECT EXECUTION OF THOSE OBJECT PROGRAMS,  
WHICH RUN PROPERLY WHEN NOT OPTIMISED. I HOPE THAT A FUTURE RELEASE WILL INCLUDE  
A NEW VERSION OF THE KIDSGROVE ALGOL SYSTEM THAT MITIGATES THE ABOVE PROBLEM.**

- The Kidsgrove system gives incorrect results for some programs containing optimised 'step-until' loops with an iteration count that exceeds 65535. A loop traversing an array would not normally have a problematic iteration count, as the largest physically possible array contains fewer than 32768 elements.
- Optimised 'for' loops evaluate the 'step' and 'until' expressions only once. This makes a difference if the body of the loop exerts a side effect on either of them: the side effect will be inoperative.
- If the body of a optimised 'step-until' loop explicitly changes the controlled variable, the change will not have its intended effect on the course of the iteration.

**THESE ARE PERMANENT LIMITATIONS OF OPTIMISED LOOPS,  
DUE TO THEIR ITERATION COUNTS BEING HELD IN THE C FIELDS OF Q STORES.**