Known Bugs in ETEC Version 1.10

Bug Identifier	Problem/Bug Description	Severity	Workaround Description	Affected Releases	Fixed Release
V1.00D-5	When the sizeof operator is applied to a constant the wrong size may result, e.g. sizeof(1) may result in "1" rather than "3".	2	Take the sizeof the desired type instead: sizeof(int)	All versions	TBD
V1.00D-7	It appears that ETEC integer promotion rules are not correct in all cases. For example, the code { unsigned char a = 1; unsigned char b = 2; int c = a-b; } should yield a value of -1 in c but instead ETEC-generated code results in 255.	2	Cases such as the example shown can be corrected through the use of explicit typecasts, e.g. int c = (int)a - (int)b;	All versions	TBD
V1.10A-1	If a 'C' function calls another 'C' function, but the top-level 'C' function is an orphan (never gets called) then this sometimes (incorrectly) triggers internal diagnostics that results in a compilation failure.	3	Commenting out such dead code avoids the problem.	All versions	V1.11A
V1.10A-2	A static array, with function scope, with size determined by an initializer, fails to compile.	3	Provide an explicit array length.	All versions	V1.11A
V1.10A-3	If a channel frame (or structure) contains a 16-bit piece of data, and two 8-bit pieces of data, packed such that the 16-bit data is at the start of a double-even address, followed by the two 8-bit items (thus making up an entire 4-byte word), then read access of the 8-bit data fails due to invalid code generation.	2	Change the data types or re-arrange the channel frame to avoid the occurrence of such data packing.	All versions	V1.20A

Bug Severity Level Descriptions:

1 – Problem causes complete work stoppage. No work-around is possible. The problem is likely to be hit by most users. This level of bug will typically trigger a new release or patch in a short time frame.

2 – A difficult problem to track down, such as incorrectly generated code. Typically there is a work-around available for this kind of bug.

3 – A bug that is easy to spot, and/or generally has a straight-forward work-around, or has minimal impact.

4 – Not truly a bug (i.e. tool is within spec.), but rather something that might affect compatibility or usability. Work-arounds available.