

Changes in low-level Ethernet code in Fugue Release
June, 1983

The Ethernet code was substantially rewritten for the Fugue release to accommodate the 10mb Ethernet and NS protocols, and to remove some (eventually all) of the Bcpl Ethernet code. The changes are largely internal, but there are some user-visible changes. For complete details, see the Lisp Ethernet documentation. Herewith are the principal visible changes:

The PUP record. PUP's are now instances of the datatype ETHERPACKET, which has several special fields and then a large data section where the actual PUP (or other level-one packet) lives. The PUP record is an overlay into this structure. The fields of the PUP record have been renamed for consistency, although the old names temporarily continue to give you time to switch over. Thus, your existing code with Fetches and Replaces of PUP fields should still work, although it absolutely MUST be recompiled. The PUPCONTENTS field is also still there, even though a PUP is now physically in one piece; this should not have any logical effect on your old code. The "extra" fields of the old PUP record no longer exist, however. There are extra fields in an ETHERPACKET that may be useful: EPFLAGS (a BYTE field) and EPUSERFIELD (a POINTER field). The PUP and other records are not included in the standard loadup; you must load the LispUsers package ETHERRECORDS.

SENDPUP -- is largely the same, except that while the pup is being sent, it is considered to be no longer in your possession, but rather in the possession of the ethernet driver. This means, in particular, that you may not write into the pup at least until it has finished transmission. A packet is in transmission while the EPTRANSMITTING flag in the packet is true. For aid in synchronizing with the transmitter, there is an additional field in the ETHERPACKET record called EPQUEUE, which says what the transmitter should do with the packet when it is finished. The value FREE means the packet should be returned to the free pool (which means you had better not still be hanging on to it after you do the SENDPUP); a value that is a SYSQUEUE means to place the packet on the specified queue. See documentation for description of SYSQUEUE's. There is an optional extra argument QUEUE to SETUPUP, to set the EPQUEUE field.

GETPUP -- takes only one argument, the pup socket, and returns a brand new pup. That is, GETPUP no longer takes a scratch pup as input and copies data into it. Similarly, EXCHANGEPUPS now ignores its INPUP argument, instead returning the pup, if any, that arrived.

CREATESOCKET, FLUSHOCKET -- are now called OPENPUPSOCKET, CLOSEPUPSOCKET. Again, the old names will exist temporarily during the switchover.

FILLPUPSOURCE -- no longer exists. If any of the 3 source fields of a pup (net, host, socket) are zero when the pup is handed to SENDPUP, the fields are defaulted appropriately.