

# Kermit-20 5.3(248)-5 Maintenance Release

Thomas DeBellis, [SLOGIN@TOMMYT.#DECnet](mailto:SLOGIN@TOMMYT.#DECnet) (TOMMYT : :SLOGIN, HECnet), 28-Nov-2023

## Table of Contents

- Summary .....2
- [236] Implement /FORKLESS connections for DECnet NRT's.....2
- [237] Implement the RETURN keyword to stop processing a TAKE file .....2
- [238] Enhanced NRT disconnect messages .....3
- [239] Use .HPTOD (High Precision Time of Day) if the monitor has it.....4
  - HPTIM% .HPTOD .....4
- [240] When logging to a disk file, display the number of bytes written .....5
- [241] Handle potential case of incorrect data being sent for CWD.....5
- [242] ECHO /TRANSACTION /DEBUG /SESSION Switches.....5
- [243] Fix off by one error which appended NUL characters in packets.....5
- [244] Fix server FINISH does not close all logs .....6
- [245] Rewrite MOVSTU to use EXTEND instead of loop .....6
- [246] Additional C escape sequences: \O and \U .....6
- [247] Rewrite IAC handling to use EXTEND instead of loop .....6
- [248] Fix incorrect string lengths being returned by C constant expansion .....7
- Kermit-20-Testing-Battery-5.3(248)-5 Updates and new Batch Tests.....7
  - K20DPD: Updated Kermit-20 Packet Decoding Example .....7
  - K20NRV: Updated Kermit-20 Server functions via Tops-20 DECnet NRT to local host .....7
  - K20NRT: Updated Kermit-20 Server functions via Tops-20 DECnet NRT to remote host.....7
  - K20DPD: Updated Kermit-20 Packet Decoding Example .....7
  - K20P8R: New Regression test against Kermit-20 4.2(174) [2-May-85] 8 bit files with parity .....7
  - K20RDC: New Communications comparison Kermit-20 4.2(174) [2-May-85] and 5.3(248)-5 .....8
  - K1036N: Kermit-10 36 bit Regression Tests via DECnet NRT.....8
- Updated Help .....8

## Summary

Version 5.3(248)-5 incorporates additional features, enhancements and fixes since the minor release of 5.3(235)-5, in June of this year (11-Jun-2023). These are as follows and can be identified in the code with the edit number as the prefix of a comment.

### [236] Implement /FORKLESS connections for DECnet NRT's

/FORKLESS may be given as a parameter on a CONNECT, SET HOST or SET LINE command when connecting to a remote DECnet host which supports Tops-10/Tops-20 NRT transport.

It specifies to use Tops-20's built-in (monitor based) NRT client instead of having Kermit create a fork and handle user communications with the remote host itself as it has historically done.

By not creating a fork with its associated memory and processor demands and by eliminating context switching for NRT traffic, /FORKLESS can save on system resources. This can result in a significantly better user experience on heavily loaded systems. For systems running billing, some charges associated with the connection will be less.

On the other hand, on a very lightly loaded system with few users, having Kermit do the terminal processing itself can sometimes yield a snappier response as two different forks are used, one to always be waiting to read user input and the other waiting for remote system output. As these are two different schedulable entities, they will wake on different conditions and can thus be scheduled more appropriately.

The use of /FORKLESS is undetectable by the remote system and is entirely separate from normal Kermit transfers. It has no effect on either the INPUT or OUTPUT commands. It is valid only for a DECnet connection and is ignored for pseudo-terminal or serial line connections.

Note that when using /FORKLESS, remember that Tops-20 itself is handling the NRT connection outside of user mode, and that therefore, Kermit-20 cannot know the remote terminal character counts and thus, these will not be displayed by SHOW LINE.

### [237] Implement the RETURN keyword to stop processing a TAKE file

RETURN immediately terminates processing of the current TAKE file and either returns processing to the TAKE file which invoked it or goes to top level. It is not an error to issue a RETURN at the terminal, but it does nothing useful other than informing you that you are not currently taking commands from a file.

The purpose of the command is multi-fold. First, in addition to judicious use of ECHO commands, stopping TAKE file processing can be a significant debugging aid. Previously, the only stop to TAKE file execution was when end of file was detected.

Second, RETURN allows information to be kept in the TAKE file which is not parsable, such as free form comments or commands which are no longer necessary.

Third, and perhaps most importantly, RETURN can also substantially speed up start up times for KERMIT .INI files which have a large number of macro definitions.

In this case, instead of having multiple separate lines defining macro names and bodies, a single `DEFINE /MAP` can be given to directly map them into memory and `.INI` processing stopped. The previous `DEFINE` statements can then follow the `RETURN` and will be ignored.

Thus, imagine the following macro definitions:

```
define foo flow-control none
define bar duplex full
define baz parity even
define boo handshake none
define CU20B set prompt "CU20B>"
    <End-of-File>
```

If `DEFINE /DUMP:KERMIT.BIN` is used to save them into binary format, then the following is going reduce the number of necessary `COMND%` parses from thirty to five, which is a reduction by a factor of six and the macros will be immediately available with no intermediate processing or conversion required.

The above may be accordingly rewritten as:

```
DEFINE /MAP KERMIT.BIN
RETURN
define foo flow-control none
define bar duplex full
define baz parity even
define boo handshake none
define CU20B set prompt "CU20B>"
    <End-of-File>
```

As the original definitions macro follow the `RETURN` statement, they will not be parsed, yet will be harder to misplace or otherwise lose.

See `HELP DEFINE` for further information on macros. Be aware that `DEFINE /COMPACT` should be used before dumping macros to use the minimum amount of disk space. See `HELP TAKE` for further information on the `TAKE` command.

## [238] Enhanced NRT disconnect messages

Native Tops-20 programs do not report errors numerically (or, in the case of other systems, not report anything at all). While all errors do have an associated number, this is never displayed to the user as a simple `JSYS` exists to do the conversion from the number to its associated string (`ERSTR%`).

Such is not the case with DECnet errors. While the error codes are symbolically defined, no associated text exists and Kermit was forced to print an arcane number. Various DECnet sources and standards were researched to come up with messages that would be understood across various platforms.

Thus, instead of printing an error code of 32 on a connection failure, Kermit will now display, "Too many connections to node" which will surely provide more clarity.

Despite this, note that some messages may require a familiarity with DECnet transport to be completely understandable. For example, error code 21 has an associated text of "Connect Initiate with illegal destination address" Since DECnet addresses may not be specified numerically under Tops-20, it indicates that the node has been defined incorrectly.

The solution here is to find a systems programmer to resolve the issue and redefine the node.

### [239] Use .HPTOD (High Precision Time of Day) if the monitor has it

This is a further enhancement to edits [207] (Properly time sub-second transfers) and [221] (Packet decoding when debugging).

By default, Tops-20 stores the time of day as fixed binary fraction of 86,400 seconds divided by  $2^{18}$  (or 262,144). This yields a resolution of one TOD tick being equal to 0.32958984375 seconds. This is perfectly fine for many uses since Tops-20 will only parse and print times with single second resolution.

A TOD tick does not have sufficient resolution for time stamping kilobaud packets and is utterly inadequate for megabaud communications and above. Edit [221] attempted to compensate for this by replicating Tops-20's internal intermediate time base to yield millisecond resolution. This had the problem that timestamps between two Kermit-20's on the same system would not always make sense, a receive appearing to be logged before the send happened.

The problem required a monitor solution and this has been done under PANDA II, which is not in general release. The HPTIM% JSYS has been modified to give a time of day with ten microsecond resolution. Kermit-20 has been modified to detect when this functionality is available and use it when logging (see below).

The 10 microsecond resolution is based on the DK10 real time clock used on the KA10 and the KI10 on which TENEX, Tops-20's predecessor was designed and results from the DK10's maximum internal frequency of 100 kilohertz. This is still adequate to time load stressed Kermit transfers and edge cases which would normally not be seen in actual usage. For example,

```
R,10-Nov-2023 17:54:09.70724, type: Y, seq: 1, len: 90, Blk: 31, Acknowledged packet type "D"  
S,10-Nov-2023 17:54:09.70780, type: D, seq: 1, len: 90, Blk: 31, Data: 23 [...]
```

In this case, the packet decoding and timing is happening on a Kermit-20 running as a server on a pseudo-terminal being communicated with another Kermit-20 on the same system. This contrived transfer is designed to stress the implementation under worst case conditions.

It can be seen that the time necessary for the Kermit-20 server to send the next data packet after having received an acknowledgement from the client for the previous one is happening in sub-millisecond time or 560 microseconds.

### HPTIM% .HPTOD

PANDA II will not be available for release for at least another calendar year. For users who wish to avail themselves of the functionality, the changes to the monitor are minimal and are included in the HPTOD subfolder and involve three monitor modules and a symbol file, viz:

Name	Protection	Pages	Bytes(Bytesize)	Write	Date and Time
------	------------	-------	-----------------	-------	---------------

APRSRV.DIF.6;P777700	3 5956(7)	12-Nov-2023 19:45:24
JSYSA.DIF.6;P777700	6 15166(7)	12-Nov-2023 19:30:11
JSYSM.DIF.10;P777700	8 19973(7)	12-Nov-2023 19:39:23
MONSYM.DIF.5;P777700	1 1147(7)	12-Nov-2023 19:27:37

Most of the text in the above files is actually documentation, analysis and comments.

## [240] When logging to a disk file, display the number of bytes written

Previously, Kermit-20 would only display the name of the file being logged to. For a number of reasons, it is useful to know how much data has been written to the log file. For example:

```
Local-Kermit-20>Show Debugging
Debugging: Packets [Decoding, 10 microsecond resolution]
Debugging log file: DECODE.LOG.4, 12050 Bytes Written, Byte Size 7
```

Note that when logging to the NUL: device, Tops-20 does not keep a count of bytes for the special destination designator (.NULIO) that Kermit-20 uses to dump the output.

As logging to NUL: is commonly only used for prototyping, debugging and profiling, using .NULIO saves a JFN slot and otherwise requires less system resources. It is also believed to be less likely to trigger a crash during periods of extreme Kermit edge case testing.

## [241] Handle potential case of incorrect data being sent for CWD

The /Decode code shows a GC packet having a great deal of NUL characters. This is an artifact of the decoding, the packet actually does have correct internal counts. However, some more care was taken to properly scrub the directory, password and atom buffers in all cases.

## [242] ECHO /TRANSACTION /DEBUG /SESSION Switches

Allow ECHO to put parsed text being displayed on the control terminal into an indicated log file. Useful for indicating exactly where a problem might be. Correct problem of final carriage return line feed not having correct parity if doing parity.

## [243] Fix off by one error which appended NUL characters in packets

In order to minimize the processor time used when SOUT%'ing to the controlling terminal or remote line or SOUTR%'ing to the network, Kermit-20 was changed to write counted strings instead of having Tops-20 do the calculation, looking for the NUL character.

The calculation of packet length mistakenly took this final NUL into account, resulting in every packet being sent with a trailing NUL, thus making for slower transfer times.

The character count between Kermit-20 4.2(174) and Kermit-20 5.2(240)-5 now differs by a single byte. This is because the older Kermit does not support large buffers whereas the newer Kermit does and an extra byte is necessary to extend the bitmask to signal this in the Initialization ("I") packet.

## [244] Fix server FINISH does not close all logs

The server finish code predated session and transaction logging, so it only closed the debugging log. Log the FINISH as a transaction (if logging transactions) and close any and all open logs.

## [245] Rewrite MOVSTU to use EXTEND MOVST instead of loop

The MOVSTU routine is used to convert a string which may possibly contain lower case letters into all uppercase. It does this by looping over strings that are long enough that using a MOVST table based approach to translate the string will save processor time. This is another example of more modern Kermit-20's trading space for time. ECHO command was updated to take the invisible /UPPERCASE switch to test MOVSTU functionality.

## [246] Additional C escape sequences: \O and \U

This is the letter O, (for 'obliterate') not the number zero (0), which would precede an octal character escape sequence. It is used to send an octal 177 (Hexadecimal FF, Decimal 127), also known as rubout or the delete character. It is sometimes necessary to erase a single character in a stream, and rubout is less likely to be eaten by intervening systems than a NUL. This is necessary for certain kinds of terminal padding.

Kermit-20 will normally eliminate NUL's (ASCII 0) from the output stream and the character cannot be reliably typed on all keyboards. Use of \U forces a NUL into the stream.

## [247] Rewrite IAC handling to use EXTEND MOVST instead of loop

Background: Telnet uses a special 8-bit character to indicate that the next byte in the terminal output stream should be interpreted as a command. This character is known as the IAC character and is octal 377, hex FF and decimal 256.

When Kermit-20 is sending binary data, it is possible that a legitimate 377 can be seen in the data stream. Further, a delete or rubout character (octal 177) sent with even parity will also get turned into an IAC after parity is applied. This latter case is perhaps very unlikely as TVT transport does not support parity.

However, since for the purposes of testing, Kermit can force parity whether the line supports it or not, the send packet routine (at `spack6:`), TRANSMIT (at `$tran6:`) and OUTPUT command (at `$outp4:`) were slightly reordered to have IAC handling be last in order to handle post parity processing.

In either an actual or artificially forced case, the IAC must be quoted (meaning doubled) in order to be transmitted properly.

IAC doubling cannot happen with a DECnet NRT transport as such signaling is done out-of-band. Nothing unusual will happen on a pseudo-terminal and this can be used for testing (see below).

Kermit-20 previously looped through each packet to determine whether IAC doubling was necessary. Rewriting it to use the EXTEND MOVST instruction is part of ongoing loop elimination and replacement, another example being found for [245].

ECHO command was updated to take the invisible /IAC switch to test IAC doubling functionality. An IAC can be forced by setting Kermit to even parity and putting \O (rubout) into the output stream, viz: ECHO "Her\oe". Be aware that you may get a TAC or inferior Telnet client deeply confused if you do this.

## [248] Fix incorrect string lengths being returned by C constant expansion

Expansion of C constants should result in the length of the resulting string being less. For example, consider the following string to be expanded "\q\t\t Hello\q" is fourteen (14) characters. Expanded, it will be " Hello", which is ten (10) characters long, a reduction in length of four (4) characters. The C expander was returning the incorrect original length of fourteen (14) for the expanded string.

## Kermit-20-Testing-Battery-5.3(248)-5 Updates and new Batch Tests

1. Rewritten Introduction of several paragraphs explaining regression testing and interpreting speed results.
2. Creation of two more batch jobs, K20P8R and K20RDC.
3. Creation of the .MIC files for sequential execution of all regular (K20ALL.MIC) and regression tests (K20175.MIC) and Kermit-10 regression tests (K10ALL.MIC).
4. Modified all Tops-20 Batch jobs for such sequential execution (via dependency counts), if set up by .MIC file.
5. Created Kermit-20-Test-Battery.xlsx to show total test processor and wall time usage.

### K20DPD: Updated Kermit-20 Packet Decoding Example

This has been modified to display byte counts for log files and the timing resolution of the log file, if packets are being logged.

### K20NRV: Updated Kermit-20 Server functions via Tops-20 DECnet NRT to local host

Fixed to follow batch standard and properly signal an error.

### K20NRT: Updated Kermit-20 Server functions via Tops-20 DECnet NRT to remote host

This has been modified to turn off flow control, increase time outs for remote DECnet sign off, show fractional wait times, display remote counts, and give an example of BYE.

### K20DPD: Updated Kermit-20 Packet Decoding Example

This has been modified to display byte counts for log files and the timing resolution of the log file, if packets are being logged.

### K20P8R: New Regression test against Kermit-20 4.2(174) [2-May-85] 8 bit files with parity

Created to test 8 bit file transfers with parity. This test case had been omitted in previous releases.

**K20RDC:** New Communications comparison Kermit-20 4.2(174) [2-May-85] and 5.3(248)-5

Created to send the same file with both Kermit-20 4.2(174) and 5.3(248)-5 and display the character counts. This was used to trouble shoot edits 243 and 248.

**K1036N:** Kermit-10 36 bit Regression Tests via DECnet NRT

Created for Kermit-20 maintenance release 5.3(235)-5 of 11-Jun-2023, which introduced 36 bit mode, yet not documented.

## Updated Help

Additional help for above enhancements, other minor corrections.