Kermit-20 5.3(255)-5 Maintenance Release

Thomas DeBellis, SLOGIN@TOMMYT.#DECnet (TOMMYT::SLOGIN, HECnet), 29-Mar-2024

Table of Contents

Summary

Version 5.3(255)-5 incorporates additional features, enhancements and fixes since the minor release of 5.3(248)-5, in November of last year (28-Nov-2023). These are as follows and can be identified in the source code with the edit number as the prefix of a comment.

The changes may roughly be classified into two areas, minor fixes to the triggering and printing of status information and handling unpunctuated directory specifications, with changes typically being implemented in multiple separate edits as functionality become more understood.

Regarding status information, Kermit did not always print the page total when requested to do so by a Control-A nor print directory headers when the directory changed in a wildcarded listing, which was important for knowing which files were being listed.

More significant was the fact that Tops-20 has very strict punctuation requirements for specifying directories, which may only be done in via what would be called an absolute path. Whereas absolute paths always start with "/" in Unix or "\" in Windows, DOS and OS/2, a Tops-20 directory is always absolute either to the entire system, beginning with a structure or absolute to the structure, the former being something like `PS:<FOO>` and the latter being `<FOO>`, assuming one is currently connected to a directory on the PS: structure.

It is critical to understand that file specifications differ *wildly*, Unix paths being incompatible with Windows, DOS or OS/2 which look nothing like MVS, Tops-20, ITS or Tops-10. As an example, the following might identify the same file or dataset or they might not:

OS	Path	Comments
Unix	`/ps/foo/bar/file.txt`	User has no idea of what the physical device might be
Windows	`p:\foo\bar\file.txt`	Single character devices, only
MVS	`PS.FOO.BAR(FILE.TXT)`	Uses partitioned datasets
Tops-20	`PS:<FOO.BAR>FILE.TXT`	Grammar uniquely identifies all parts of specification
ITS	`FOOBAR;FILE TXT`	ITS has no idea of subdirectories
Tops-10	`PS:FILE.TXT[FOO.BAR]`	Assumes symbolic paths, not PPN's

The problem for a Kermit client is that various operating systems define these punctuation characters to do different things, so specifying a remote file specification can be difficult. For example, on Unix, Windows, DOS, and OS/2, the "<" and ">" characters are usurped by the shell for input and output redirection respectively while Tops-20 uses semicolon (";") as either a comment character or to specify a file attribute.

In an attempt to ease the problem, Tops-20 Kermit will now accept relative directory specifications in two forms. First, a subdirectory in the currently connected directory may simply be given by name and second, Unix, Windows, DOS, and OS/2 superior directory syntax has been implemented by a new Kermit-20 generic command "G", meaning CDUP, which will be detailed, below.

A new batch test has been created to validate this and help documentation has been updated.

[249] Implement relative directory connect

A remote Kermit-20 running in server mode now supports 'relative' or syntax-free connects to subdirectories of the currently connected directory.

For example, consider the case of a directory named FOO on the PS: structure, which has two subdirectories, BAR and BAZ. The first directory (known as a top-level directory) would be specified as PS:<FOO>, while the latter two subdirectories would be specified as PS:<FOO.BAR> and PS:<FOO.BAZ>, respectively.

All three subdirectories must normally be specified with complete syntax, which can be difficult for remote parsers to grasp, the result being that input and output may be silently redirected by certain local clients.

Assuming again that remote Kermit-20 is currently connected to PS:<FOO>, specifying "CWD BAR" will get you into PS:<FOO.BAR>, while "CWD BAZ" will connect to PS:<FOO.BAZ>. Be aware that such relative directories specifications do not exist in standard Tops-20 and that Kermit-20 is silently doing the conversion. Thus, it is marginally computationally faster to use the Tops-20 native form of the directory specification than relative directory specifications.

On PANDA and Stanford based Tops-20 monitors, the user may specify "CWD <.BAZ>" and "CWD <.BAZ>" to perform the same functions to GTJFN% and COMND%. While this works, JFNS% does not print the full absolute directory specification, which can confuse programs which display this (such as Emacs) or depend on it (such as DUMPER).

Be aware that CWD is a Unix term and that other operating systems may have subtle to vastly different understandings of what a 'working' directory means. Kermit-20's HELP CWD should be reviewed for an overview of the Tops-20 nuances.

[251] Emit new directory name when either wildcarded structure or directory changes

When doing a wildcarded directory listing, Kermit-20 would give no indication of which files were in which directory while stepping the wildcarded file or directory specification, either when a new sub-directory had been hit or when a directory.

[252] Fix some of the directory parsing logic to properly handle .CMDEV and .CMDIR

Tops-20 implements the idea of a logical device, which simply looks like a structure specification. This is internally expanded by Tops-20 to a full file or directory specification. For example, suppose directory PS:<FOO> has been defined as logical name BAR:. "CWD PS:<FOO>" and "CWD BAR:" would get you to the same place.

Logical devices therefore introduce lexical ambiguity, making it also possible to attempt to connect to devices which are not structured, such as PTY:, TTY: and DCN:. The change allows Kermit-20 to efficiently detect such situations with more granularity and to silently convert a ‘bare’ directory specification (viz, “PS:<FOO>”) to a wildcarded file specification (“PS:<FOO>*.*”), where appropriate.

“NUL:” is special cased to do nothing, which is used for testing and debugging.

[253] Fix ^A to print the number of pages sent and the total; was always omitting total

While Control-A (^A) would display the number of pages being sent, there was no way to determine how far the transfer had progressed. ^A now displays the total being sent so the user can gauge the progression. For example:

```
Kermit-20>SEND DOC:JSYS_REFERENCE.MEM.0 (as) "NUL:"
^A for status report, ^X to cancel file, ^Z to cancel batch.
TOMMYT:<DOCUMENTATION>JSYS_REFERENCE.MEM.1 as NUL:
.......................................
^A
Sending TOMMYT:<DOCUMENTATION>JSYS_REFERENCE.MEM.1, file bytesize 7, i/o
bytesize 7
 (compression) (block check type 1)
 At page 21 of 701
 Files: 0, packets: 1529, chars: 53221
 NAKs: 0, timeouts: 0
........................
```

[254] Implement CDUP, generic 'G'

CDUP is named after the FTP server verb which performs the same function. However, since Kermit-20 does not use verbs, CDUP is implemented by a previously unused Kermit protocol generic letter, “G”.

CDUP requests the remote Kermit to change the remote working directory to the next higher directory in the file system hierarchy. Like CWD (generic “C”) is the intended effect is that this will become the default area that the remote system will attempt to create or find files, absent an explicit specification of the area.

The difference between the two commands is that no parameter is given to CDUP and that it is the responsibility of the remote Kermit server to figure out what directory to connect to and thence attempt the connect. Here is an example of CDUP usage against a Tops-20 Kermit in server mode:

```
      Kermit-20>REMOTE PWD
      TOMMYT:<OINKY.HOLMES.MEMOIRS>
      Kermit-20>REMOTE CDUP
      TOMMYT:<OINKY.HOLMES>
      Kermit-20>REMOTE CDUP
      TOMMYT:<OINKY>
```

Giving a CDUP while connected to a top-level directory is not considered an error, yet performs no useful function, the user remaining connected to the same directory.

Be aware that CDUP is a Unix term and that other operating systems may have subtle to vastly different understandings of what changing a changing a working directory means. In particular, since CDUP takes no parameters, it assumes that access to the superior directory will always work, meaning no password will ever be required.

This is true for a PANDA or Stanford based monitor running super-domestic structures, but may not be true for a standard DEC Tops-20 monitor. Kermit-20's `HELP CWD` should be reviewed for an overview of the Tops-20 nuances.

[255] Make CWD accept ".." as token for Unix, DOS, Windows, and OS/2 CDUP

Since not all Kermit clients know about generic "G" (CDUP)., a Kermit-20 running in server mode will accept a parameter of "`..`" to CWD as a special case token meaning to do a CDUP. Kermit-20 will ignore any password and silently convert the command into a CDUP. Note that is a special explicit case internal to Kermit-20 only and that otherwise, the bare sequence of "`..`" is an erroneous Tops-20 directory specification.

Kermit-20-Testing-Battery-5.3(255)-5 Updates and new Batch Test

1. Creation of a new batch job, `K20GPT` to test CDUP, CWD `..`, and relative directory connects.
2. Updated Introduction to include CDUP.
3. Updated Triggered Sequential Test to include new batch job, `K20GPT` and to more clearly state certain measurement assumptions.
4. Small updates to Speed Measurement Considerations.
5. Update of the `K20ALL.MIC` file to include `K20GPT`.
6. Update batch jobs which perform regression tests against Kermit-20 4.2(174) [2-May-85] to handle certain incompatibilities with newer KERMIT.INI files.

`K20GPT`: Created to test `CDUP`, `CWD ..`, and relative directory connects.

In addition to packet logging of the generic "G" sequence, the following extracts are of relevance:

Example of local `CDUP`

```
21:43:45 USER   Local-Kermit-20>*Local CWD STAR:<FTP.1.2.3.4>
21:43:45 USER   [Connected to STAR:<FTP.1.2.3.4>]
21:43:45 USER   Local-Kermit-20>*Local CDUP
21:43:45 USER   [Connected to STAR:<FTP.1.2.3>]
21:43:45 USER   Local-Kermit-20>*Local CDUP
21:43:45 USER   [Connected to STAR:<FTP.1.2>]
21:43:45 USER   Local-Kermit-20>*Local CDUP
21:43:45 USER   [Connected to STAR:<FTP.1>]
21:43:45 USER   Local-Kermit-20>*Local CDUP
21:43:45 USER   [Connected to STAR:<FTP>]
21:43:45 USER   Local-Kermit-20>*Echo "Last CDUP at Top-level does nothing"
21:43:45 USER   Last CDUP at Top-level does nothing
```

```
21:43:45 USER
21:43:45 USER   Local-Kermit-20>*Local CDUP
21:43:45 USER   [Remaining connected to top-level directory STAR:<FTP>]
```

Example of local `CWD ..`

```
21:43:45 USER   Local-Kermit-20>*Local CWD STAR:<FTP.1.2.3.4>
21:43:45 USER   [Connected to STAR:<FTP.1.2.3.4>]
21:43:45 USER   Local-Kermit-20>*Local CWD ..
21:43:45 USER   [Connected to STAR:<FTP.1.2.3>]
21:43:45 USER   Local-Kermit-20>*Local CWD ..
21:43:45 USER   [Connected to STAR:<FTP.1.2>]
21:43:45 USER   Local-Kermit-20>*Local CWD ..
21:43:45 USER   [Connected to STAR:<FTP.1>]
21:43:45 USER   Local-Kermit-20>*Local CWD ..
21:43:45 USER   [Connected to STAR:<FTP>]
21:43:45 USER   Local-Kermit-20>*Echo "Last CWD at Top-level does nothing"
21:43:45 USER   Last CWD at Top-level does nothing
21:43:45 USER
21:43:45 USER   Local-Kermit-20>*Local CWD ..
[Remaining connected to top-level directory STAR:<FTP>]
```

Example of remote relative directory connecting

```
21:43:45 USER   Local-Kermit-20>*Remote CWD "PS:<OINKY>" ""
21:43:45 USER   TOMMYT:<OINKY>
21:43:45 USER   Local-Kermit-20>*Remote CWD "HOLMES" ""
21:43:45 USER   TOMMYT:<OINKY.HOLMES>
21:43:45 USER   Local-Kermit-20>*Remote CWD "MEMOIRS" ""
21:43:45 USER   TOMMYT:<OINKY.HOLMES.MEMOIRS>
```

Example of remote `CDUP` from directory `TOMMYT:<OINKY.HOLMES.MEMOIRS>`

```
21:43:45 USER   Local-Kermit-20>*Remote PWD
21:43:45 USER   TOMMYT:<OINKY.HOLMES.MEMOIRS>
21:43:45 USER   Local-Kermit-20>*Remote CDUP
21:43:45 USER   TOMMYT:<OINKY.HOLMES>
21:43:45 USER   Local-Kermit-20>*Remote CDUP
21:43:45 USER   TOMMYT:<OINKY>
21:43:45 USER   Local-Kermit-20>*Echo "Last CDUP at Top-level does nothing"
21:43:45 USER   Last CDUP at Top-level does nothing
21:43:45 USER
21:43:45 USER   Local-Kermit-20>*Remote CDUP
21:43:45 USER   TOMMYT:<OINKY>
```

Example of remote absolute directory connecting

```
15:51:51 USER   Local-Kermit-20>*Remote CWD "PS:<OINKY.HOLMES.MEMOIRS>" ""
15:51:51 USER   TOMMYT:<OINKY.HOLMES.MEMOIRS>
```

Example of remote `CWD ..`

```
15:51:51 USER   Local-Kermit-20>*Remote PWD
15:51:51 USER   TOMMYT:<OINKY.HOLMES.MEMOIRS>
```

```
15:51:51 USER   Local-Kermit-20>*Remote CWD ".." ""
15:51:51 USER   TOMMYT:<OINKY.HOLMES>
15:51:51 USER   Local-Kermit-20>*Remote CWD ".." ""
15:51:51 USER   TOMMYT:<OINKY>
15:51:51 USER   Local-Kermit-20>*Echo "Last CWD .. at Top-level does nothing"
15:51:51 USER   Last CWD .. at Top-level does nothing
15:51:51 USER
15:51:51 USER   Local-Kermit-20>*Remote CWD ".." ""
15:51:51 USER   TOMMYT:<OINKY>
```

K20NOL, K20POL, K20PNW, K20POR, K20P8R, K20RDC: Enhanced Error Screening

These are all batch jobs that were designed to perform regression tests against Kermit-20 4.2(174) [2-May-85]. However since Kermit-20 version 4.2(174) and Kermit-20 version 5.3(255)-5 read the same KERMIT.INI file in the user's login directory, the older Kermit-20 will issue errors because of newer commands, new parameters and parameters with larger digital dynamic range.

Kermit-20 4.2(174) would consequently issue errors and these would cause the batch job to be aborted. Previously, these were handled manually by removing the incompatibilities from the KERMIT.INI file, a tedious an error prone activity. Now, BATCON (the batch controller) is instructed to ignore errors during Kermit-20 4.2(174) start up, viz:

```
22:40:24 BATCH  @NOERROR
22:40:24 USER   *Input  "\r\nKermit-20>"
22:40:24 USER   ?Null switch or keyword given - define /map h:kermit.bin
22:40:24 USER   ?Does not match switch or keyword - return
22:40:24 USER   ?Null switch or keyword given - define "TOMMYT" prompt
                 "TOMMYT::Kermit-20>"
22:40:24 USER   ?Null switch or keyword given - define "VENTI2" prompt
                 "VENTI2::Kermit-20>"
22:40:24 USER   ?Null switch or keyword given - define "Tops-20" flow
                 none,handshake none,send packet-length 40,receive packet-
                 length 40,server 0,block-check 2,parity none,delay 20,send
                 timeout 25,receive timeout 24,retry initial-connection
                 30,retry packets 35,ITS-binary off,file bytesize 36
22:40:24 USER   ?Comma not given - define STRANGE-DELAY delay 20.1259,send
                 timeout 14.5679,receive timeout 13.2222,retry packets
                 10,handshake none,send pause 0.150,receive pause
                 0.200,receive server-timeout 15.0123
22:40:24 USER   ?Illegal packet size?
22:40:24 USER   ?Illegal packet size?
22:40:24 USER   ?Illegal packet size?
22:40:24 USER   ?Illegal packet size?
22:40:24 USER   TOPS-20 KERMIT version 4.2(256)
22:40:24 USER
22:40:24 USER   Kermit-20>
22:40:24 USER   Local-Kermit-20>
22:40:24 BATCH  @ERROR
```

Batch error detection is shut off with `@NOERROR` and turned back on with `@ERROR`. These errors are actually benign in the sense that they are not being tested against.

Updated Help

Additional help for above enhancements, other minor corrections.