

TaskMaster / TMLite Quick Reference

Batch Commands / Scripting Language

(As of: v4.18 / Updated: 15 January 2008)

Enhanced Batch Commands / Script Language supported in Task (.TSK) scripts
run manually (TMRUN), scheduled (TMSCHEDULE), or launched (CALL)

Syntax : { } item required [] item optional | OR Options : Must be separated by space

Commands

Command Syntax	Comments / Examples
ABORT [cmd task]	Cancel active task (ABORT) or the specified cmd (ABORT cmd) or task (ABORT task). (See also ABORT Local Server Command in the Extended Console Commands.)
ACTIVE SCREEN ["screen_name"]	Note: Obsolete (supported for backwards compatibility). Replacement: CONSOLE SCREEN
CALC {Dynamic Variable} {calculation}	Perform basic integer math calculations (+ Add, - Subtract, * Multiply, / Divide) storing result in specified Dynamic Variable. // Results stored in %0, %VAR00% & %PCT%, respectively // Note: %PCT% is assumed to have been previously assigned using VARALIAS. CALC %0 (((100 - %1) * %2) / %3) + 4 CALC %VAR00% 100 - %CPU_UTIL% CALC %PCT% (%VOL_USED% * 100) / %VOL_SIZE% <i>Notes:</i> Evaluates Left-to-Right. Entire calculation must reside on one line. If Dynamic or Environment Variables are used in the calculation, the current contents are substituted prior to computation. If result variable has fixed digit format, CALC will retain format (refer to DEFINE).
CALL [[vol:]path\]{task}[.ext]	Calls (launches) specified {task}. Task issuing CALL is paused until CALLED {task} completes.
CHANGE SCREEN ["screen_name"]	Note: Obsolete (supported for backwards compatibility). Replacement: CONSOLE SCREEN
CLOSE READ [#n] CLOSE WRITE [#n] CLOSE	CLOSE READ closes active OPEN READ files (all or #0 - #9 as specified) CLOSE WRITE closes active OPEN WRITE files (all or #0 - #9 as specified) CLOSE all active OPEN READ & OPEN WRITE files
CONSOLE SCREEN ["screen_name"]	Changes and activates the screen that is actively displayed on the Server Console. Notes: "screen_name" can be partial (i.e., "System" for the System Console). Specifying "" or nothing activates the next NLM screen. Also sets CURRENT SCREEN to the new screen.
CURRENT SCREEN ["screen_name"]	Changes the screen against which the active task performs keyin / screen interrogation commands. Notes: "screen_name" can be partial (i.e., "System" for the System Console). Specifying "" or nothing activates the next NLM screen.
DEBUG {OFF ON}	Disables (OFF) / Enables (ON) the logging of task processing activity to a file with the same vol:path\filename as the task but with a .DBG extension (default = OFF). <i>Note:</i> Only practical for task debugging.
DEFINE {Dynamic Variable} {string}	Define the contents of a Dynamic Variable (Aliased or Defined). // Store text string 'Hello World!' in Defined Variable %0 DEFINE %0 Hello World!

```
// Store custom greeting (with day/date) in Defined Variable %
VAR00%
DEFINE %VAR00% Today is %DAY_OF_WEEK%, %
MONTH_NAME% %DAY%, %YEAR%.

// Store pre-formatted 4 digit value of 0000 in Defined Variable
%0
DEFINE %0 0000

// Increment value of Defined Variable %0 by 2 (0000 + 2 =
0002) and store result in Defined Variable %1
DEFINE %1 %0+=2

// Decrement value of Defined Variable %1 by 1 (0002 - 1 =
0001) and store result in Aliased Variable %COUNTER%
DEFINE %COUNTER% %1-=1
```

Notes: `string` can be a text string, a Dynamic or an Environment Variable, or any combination thereof, up to 511 bytes in length. `string` is stored exactly as specified, after all Variables have had their contents substituted, including any quotes or trailing comments. Increment (`+=`) / Decrement (`-=`) only supported for Dynamic Variables which contain a valid numeric string.

DELAY {##}

Silently suspend batch processing for {##} milliseconds (min = 50 ms - max = 4294967295 ms).

```
ECHO [. | string]
ECHO [OFF | ON]
ECHO_WRITE [#n] [. | string]
```

Echo (display) any trailing comments to screen created for the task.

```
// Echo a blank line
ECHO.

// Echo the day of the week (Monday - Sunday)
ECHO Today is %DAY_OF_WEEK%!

// Echo the batch commands as processed (default = OFF)
ECHO ON

// Echo a blank line and write it to the OPEN_WRITE #0 file
ECHO_WRITE.

// Echo the day of the week and write it to the OPEN_WRITE
#3 file
ECHO_WRITE #3 Today is %DAY_OF_WEEK%!
```

Notes: ECHO OFF | ON controls the display of Batch Commands to the screen created for the active task (default = OFF). `string` can be a text string, a Dynamic or an Environment Variable, or any combination thereof. `string` is stored exactly as specified, after all Variables have had their contents substituted, including any quotes or trailing comments. ECHO WRITE both displays (echoes) the trailing comments and writes the information to a previously opened OPEN WRITE [#n] file. Failure to properly open an OPEN WRITE [#n] file prior to using ECHO_WRITE will generate a batch processing error.

EXIT

Exits (silently terminates) the currently active task.

FINDCHR {Dynamic Variable} {"c"} {"str"} [opt]

Find the position of character ("c") in string ("str") and store the result in the specified Aliased or Defined Variable.

	<p><i>Notes:</i> The first specification receives the result of the search (0 = not found or 1 - length of string = position of "c" in "str") so it must be a Aliased or Defined Variable. The second and third can be a text string, a Dynamic an Environment Variable, or any combination thereof but must be enclosed in double quotes. If the second specification contains more than one character, only the first character is used for the search.</p> <p>[opt]: CASE option for case sensitive search &/or LAST option to match the last occurrence of "c" in "str".</p>
FINDLEN {Dynamic Variable} {"str"}	<p>Find the length of (number of characters in) string ("str") and store the result in the specified Aliased or Defined Variable.</p> <p><i>Notes:</i> The first specification receives the result (the length) so it must be an Aliased or Defined Variable. The second can be a text string, a Dynamic or an Environment Variable, or any combination thereof but must be enclosed in double quotes.</p>
FINDSTR {Dynamic Variable} {"sub"} {"str"} [opt]	<p>Find the position of substring ("sub") in string ("str") and store the result in the specified Aliased or Defined Variable.</p> <p><i>Notes:</i> The first specification receives the result of the search (0 = not found or 1 - length of string = position of "sub" in "str") so it must be an Aliased or a Defined Variable. The second and third can be a text string, a Dynamic or an Environment Variable, or any combination thereof but must be enclosed in double quotes.</p> <p>[opt]: CASE option for case sensitive search &/or LAST option to match the last occurrence of "sub" in "str".</p>
GOTO {label}	<p>Transfers batch execution to the line marked by {label}.</p> <pre>// Jump to the label :EXIT GOTO EXIT</pre> <pre>// Conditionally jump to the label :EXIT IF ERRORLEVEL GOTO EXIT</pre> <p><i>Notes:</i> GOTO should not be used within IF/WHILE structures. Any task line beginning with a colon (:) is considered a label.</p>
KEYIN {key} KEYIN {"string"} KEYIN {key "string"} [{key "string"}] [...]	<p>Used to enter string and/or special keys into the screen against which the active task performs keyin / screen interrogation commands as if input from the Server's keyboard.</p> <pre>// Type the "password" then press ENTER KEYIN "password" ENTER</pre> <p>Special keys: BACKSPACE, ENTER, ESC, ESCAPE, F1 ... F12, UP, DOWN, LEFT, RIGHT, PGUP, PGDN, HOME, END, INS, INSERT, DEL, DELETE, SHIFT-key, ALT-key, CTRL-key, ALT-A ... ALT-Z, CTRL-A ... CTRL-Z</p> <p><i>Notes:</i> Multiple "string" & key combinations can be used in a single KEYIN. "string" can be a text string, a Dynamic or an Environment Variable, or any combination thereof but must be enclosed in double quotes. Use CONSOLE SCREEN / CURRENT SCREEN to change the current task I/O screen.</p>
OPEN READ [#n] [[vol:]path\]{file}	<p>Opens the specified text file ([[vol:]path\]{file})) for sequential read input of text data using the READ Batch Command.</p> <p><i>Notes:</i> The ERRORLEVEL condition is set TRUE if the file cannot be opened. Up to 10 (#0 - #9) OPEN READ files can be</p>

	<p>open simultaneously. If no handle number (#0 - #9) is specified, handle #0 is used. Issuing an OPEN READ using a handle already in use will result in the previously opened file being closed before an attempt is made to open the new file.</p>
OPEN WRITE [#n] [[vol:]path\]{file} [opt]	<p>Opens the specified text file ([[vol:]path\]{file}) for sequential write output of text data using the ECHO WRITE or WRITE Batch Commands.</p> <p><i>Notes:</i> The ERRORLEVEL condition is set TRUE if the file cannot be opened. If no handle number (#0 - #9) is specified, handle #0 is used. Up to 10 (#0 - #9) OPEN WRITE files can be open simultaneously. Issuing an OPEN WRITE using a handle already in use will result in the previously opened file being closed before an attempt is made to open the new file. [opt]: Specifying the TRUNC or TRUNCATE option causes the file to be truncated (overwritten, default = append).</p>
PARSE {Dynamic Variable} {string} {#-#[,..]} [opt]	<p>Parse (extract) data from string into an Aliased or Defined Variable.</p> <pre>// Parse byte 80 of Defined Variable %2 into Defined Variable %1 PARSE %1 %2 80</pre> <pre>// Parse bytes 1-10 of Defined Variable %2 into Defined Variable %1 PARSE %1 %2 1-10</pre> <pre>// Parse byte 10 and bytes 20-29 of Defined Variable %2 into Defined Variable %VAR01% PARSE %VAR01% %2 10,20-29</pre> <pre>// Parse byte 10 and bytes 20-29 of Defined Variable %2 into Defined Variable %VAR01% appending to any existing contents PARSE %VAR01% %2 10,20-29 APPEND</pre> <pre>// Parse bytes 1-10 and bytes 40-99 of Defined Variable %2 into Aliased Variable %KEY% (truncating if shorter) PARSE %KEY% %2 1-10,40-99 PACK</pre> <p><i>Notes:</i> string can be a text string, a Dynamic or an Environment Variable, or any combination thereof. Up to 10 byte ranges can be specified. [opt]: If specified amount of data to PARSE is more than available, extra bytes are padded (space filled) unless PACK option specified. Parsed data overwrites contents of destination unless APPEND option is specified.</p>
PAUSE [##]	<p>Suspend batch processing until a key is pressed on the Server's keyboard of the specified (##) timeout period occurs (in seconds).</p>
READ [#n] {Dynamic Variable} [#-#[,..]] [opt]	<p>Reads the next sequential text record from the OPEN READ [#n] file and parses / stores the data into the specified Aliased or Defined Variable.</p> <pre>// Read next sequential record and store data in Defined Variable %1. READ %1</pre> <pre>// Read next sequential text record and store bytes 10-20 in Defined Variable %1.</pre>

```
READ %1 10-20
```

```
// Read next record, storing bytes 1, 11-50 and 512-598 in  
Aliased Variable %DATA% (pad with spaces if less than 128  
bytes retrieved).
```

```
READ %DATA% 1,11-50,512-598
```

```
// Read next record, storing bytes 1 and 10-127 in Aliased  
Variable %DATA1% (pack or truncate if less than 119 bytes  
retrieved).
```

```
// Using same record, store bytes 1 and 128-254 in Aliased  
Variable %DATA2% (pad with spaces if less than 128 bytes  
retrieved).
```

```
READ %DATA1% 1,10-127 PACK %DATA2% 1,128-254
```

Notes: The ERRORLEVEL condition is set TRUE if a READ error occurs or once the End Of File (EOF) has been reached. #n (#0 - #9) associates the READ to the corresponding OPEN READ #n file. Although the full text record is processed (up to the CR/LF End Of Record or 4095 bytes, whichever is less), a maximum of 511 bytes can be transferred into a single variable (maximum supported variable length). Multiple variable / key combinations are supported.

[opt]: If specified amount of data to READ is more than available, extra bytes are padded (space filled) unless PACK option specified.

REFORMAT {Dynamic Variable} [opt]

Reformat the data stored in the specified Aliased or Defined Variable.

```
// Left justify (shift data left to replace leading spaces) the data  
in Defined Variable %1 (Before: ' 01210 ' / After: '01210  
' )
```

```
REFORMAT %1 LEFT
```

```
// Right justify (shift data right to replace trailing spaces) the  
data in Defined Variable %1 (Before: ' 01210 ' / After: '  
01210 ')
```

```
REFORMAT %1 RIGHT
```

```
// Pack (remove leading zeroes / leading & trailing spaces) the  
data in Defined Variable %1 (Before: ' 01210 ' / After:  
'1210 ')
```

```
REFORMAT %1 PACK
```

```
// Re-size to 6 bytes the data in Defined Variable %1 (Before: '  
01210 ' / After: ' 01210 ')
```

```
REFORMAT %1 6
```

```
// Left justify, pack and re-size to 6 bytes the data in Defined  
Variable %VAR00% (Before: ' 01210 ' / After: '1210 ')
```

```
REFORMAT %VAR00% 6 LEFT PACK
```

```
// Pack, right justify and re-size to 6 bytes the numeric data in  
Aliased Variable %COUNT% (Before: ' 01210 ' / After: '  
1210 ')
```

```
REFORMAT %COUNT% 6 PACK RIGHT
```

```
// Pack, right justify and re-size to 6 bytes the text data in  
Aliased Variable %COUNT% (Before: ' 01210 ' / After: '  
01210 ')
```

```
REFORMAT %COUNT% 6 PACK RIGHT TEXT
```

	<p><i>Notes:</i> The single quote marks in the examples are only used to delineate the data (i.e., not considered part of the variable contents).</p> <p>[opt]: LEFT shifts the data to the left to replace any leading spaces. RIGHT shifts the data to the right to replace any trailing spaces. PACK strips leading zeroes for numeric data (unless the TEXT option is specified), strips leading and trailing spaces for alphanumeric data. Re-sizing truncates longer data and pads (space fills) shorter data.</p>
REPLACE {Dynamic Variable} {"??[..]"}	<p>Searches the specified Aliased or Defined Variable for the first character in each specified pair and replaces every occurrence found with the second character in each specified pair.</p> <p>// Replace dot with dash in Defined Variable %1 (Before: "1.2.3" / After: "1-2-3") REPLACE %1 ".-"</p> <p>// Replace braces with brackets in Aliased Variable % GREETING% (Before: "{Hi}" / After: "[Hi]") REPLACE %GREETING% "[]"</p> <p><i>Note:</i> Replacement string must "??" have at least one pair of characters specified.</p>
REWIND {READ WRITE} [#n]	Reset sequential file record pointer to start of file (same as if closed then re-opened).
SAVE SCREEN [[vol:]path\]{file} [opt]	<p>Captures the text image of the the current screen against which the active task performs keyin / screen interrogation commands to {file}.</p> <p>[opt]: TXT TEXT option writes character data only (default). BIN BINARY option writes character data with screen attributes. Use CONSOLE SCREEN / CURRENT SCREEN to change the current task I/O screen.</p>
SHIFT	Shifts the contents of the Defined Variables %0 - %9 left one position, clearing %9.
SHUTDOWN	Equivalent to NetWare Server Console DOWN command. <i>Note:</i> Any open files on the Server are automatically closed without prompting.
SLEEP {##}	Silently suspend batch processing for {##} seconds.
SYSENV GET {Dynamic Variable} {string} SYSENV PUT {string}	<p>GET retrieves the value for the NetWare System Environment variable matching {string} and stores it in (Dynamic Variable). PUT stores {string} in the NetWare System Environment table.</p> <p><i>Notes:</i> string can be a text string, a Dynamic or an Environment Variable, or any combination thereof. GET {string} should be the desired NetWare System Environment variable, excluding the trailing equal sign (i.e., PATH not PATH=). SET {string} should be the full variable to be stored in the NetWare System Environment table (i.e., PATH=SYS:\SYSTEM). NetWare System Environment variables and TaskMaster Environment Variables are separate entities. (<i>NetWare uses a single preceding % designator while TaskMaster encloses the variable name with a % on each end.</i>) They are not interchangeable even if they share the same name.</p>
TOCALENDAR {Dynamic Variable} {day}	Converts numeric day (%NDAY_OF_YEAR% or 1 - 366 or 001 - 366) to corresponding calendar date format (yyyymmdd).

```
// Convert first day of year to calendar and store in Defined
Variable %0 (After: %0 = 20040101) during 2004
TOCALENDAR %0 1

// Convert 60th day of year to calendar and store in Defined
Variable %1 (After: %1 = 20040229) during 2004
TOCALENDAR %1 60

// Set Defined Variable %2 to today's numeric day of year plus 7
days
// then convert that to a calendar format and store the result in %
3
DEFINE %2 %NDAY_OF_YEAR%+=7
TOCALENDAR %3 %2
```

Note: day can be a text string, a Dynamic or an Environment Variable, or any combination thereof. The day number can be one to three digits (1 - 366 or 001 - 366). If a year is not specified (1 - 366 or 001 - 366 only), the current year is assumed.

TODAYOFYEAR {Dynamic Variable} {date}

Converts calendar date (yyyymmdd or mmdd) to corresponding numeric day (%NDAY_OF_YEAR%).

```
// Convert 14 February (any year) to numeric day of year then
store the result in Defined Variable %0 (After: %0 = 045)
TODAYOFYEAR %0 0214

// Convert 01 March 2003 to numeric day of year then store the
result in Defined Variable %1 (After: %1 = 060)
TODAYOFYEAR %1 20030301

// Convert 01 March 2004 to numeric day of year then store the
result in Defined Variable %2 (After: %2 = 061) {2004 was a
leap year}
TODAYOFYEAR %2 20040301
```

Note: date can be a text string, a Dynamic or an Environment Variable, or any combination thereof. Calendar date must be formatted either as mmdd (0615) or yyyymmdd (20040615). If a year is not specified (mmdd only), the current year is assumed.

TOLOWER {Dynamic Variable}

Convert any alpha chars in the specified Dynamic Variable to lower case.

TOUPPER {Dynamic Variable}

Convert any alpha chars in the specified Dynamic Variable to upper case.

VARALIAS {Dynamic Variable} {alias}

Assign an alias (rename) a defined Dynamic Variable (i.e., create an alias reference to the variable).

```
// Assign the alias name %COUNTER% to the reserved
Dynamic Variable %0
// Note: Reserved name %0 remains valid - %VAR00%
becomes %COUNTER%
VARALIAS %0 %COUNTER%
```

```
// Assign an alias name (%NUMBER_9%) to replace the default
defined Dynamic Variable name %VAR09%
// Note: Reserved name %9 remains valid - %VAR09%
becomes %NUMBER_9%
VARALIAS %VAR09% %NUMBER_9%
```

// Rename the Alias Variable %NUMBER_9% to %

```

NUMBER_NINE%
// Note: Reserved name %9 remains valid - %NUMBER_9%
becomes %NUMBER_NINE%
VARALIAS %NUMBER_9% %NUMBER_NINE%

// Restore default name %VAR09% to the Aliased Variable %
NUMBER_NINE%
// Note: Reserved %9 remains valid - Default name %VAR09%
restored
VARALIAS %NUMBER_NINE% %VAR09%

Notes: %VAR00% - %VAR99% are default names assigned to
the 100 Defined Variables available to each task, each of which
can be aliased or renamed to more appropriate references. The
first specification must be a valid a Dynamic Variable. The
second specification must be a valid and unique alias name.
Renaming (or aliasing) a variable does not affect its contents.
Once aliased, the previous name is not longer valid (except for
the reserved names %0-%9). Naming rules: Must begin and end
with a percent sign (%); First character after initial percent sign
(%) cannot be numeric (0-9); Maximum of 15 characters,
including percent signs (%); Alias names cannot contain spaces.

```

WAIT {##}

Suspend batch processing for {##} seconds with countdown message.

WRITE [opt] [#n] [string]

Writes string to the OPEN WRITE [#n] file including trailing EOL (End Of Line - CR/LF).

Notes: The ERRORLEVEL condition is set TRUE if a WRITE error occurs. #n (#0 - #9) associates the WRITE to the corresponding OPEN WRITE #n file. string can be a text string, a Dynamic or an Environment Variable, or any combination thereof. string is stored exactly as specified, after all Variables have had their contents substituted, including any quotes or trailing comments.

[opt]: -EOL suppresses the End Of Line (CR/LF) normally appended to each WRITE, allowing multiple WRITE commands to be issued to build a single contiguous text record.

Conditional Structures

Used with Conditional Tests to direct the logic flow for intelligent processing.

Command Syntax

```

IF [NOT] {cond} [GOTO {label}]
IF [NOT] {cond} [THEN command...]

IF [NOT] {cond} [THEN BEGIN]
    command(s)...
ENDIF

IF [NOT] {cond}
    command(s)...
ENDIF

IF [NOT] {cond} [[AND|OR] [NOT] {cond}]
    command(s)...
[ELSEIF [NOT] {cond}
    command(s)...]
[ELSE
    command(s)...]
ENDIF

```

Comments / Examples

```

// Check if current task (System Environment Variable %
TASK%) is already active
IF ACTIVE_TASK %TASK% THEN EXIT

// Check if today is New Years Day (01/01 - System
Environment Variables %MONTH% & %DAY%)
IF %MONTH%==01 AND %DAY%==01 THEN ECHO Today is:
New Years!

// Send hourly messages to employees (nested IF structyre
example)
IF %MINUTE%==00
    IF NOT %HOUR24%>07 OR NOT %HOUR24%<18
        // Not 8:00am - 5:00pm so skip.
    ELSEIF %HOUR24%==12
        SEND "*Time to Eat*"
    ELSEIF %HOUR24%==17
        SEND "*** Go Home ***"
    ELSE

```

```
SEND "Back to Work!"
ENDIF
ENDIF
```

Notes: IF / ELSEIF conditions are evaluated sequentially (in order). Supports multiple AND|OR logic (cannot mix AND/OR) & multiple ELSEIFs. Variables used for cond can be a text string, a Dynamic or an Environment Variable, or any combination thereof.

```
WHILE
  command(s)...
  [IF [NOT] {cond} THEN] BREAK
  [IF [NOT] {cond} THEN] CONTINUE
LOOP
```

```
WHILE [NOT] {cond}
  command(s)...
  [IF [NOT] {cond} THEN] BREAK
  [IF [NOT] {cond} THEN] CONTINUE
LOOP
```

```
// Set (DEFINE) Defined Variable %0 to 00 (zero - 2 fixed digit size)
DEFINE %0 00
```

```
// Increment Defined Variable %0 (0) by 1 until 10 (00 - 10 retains 2 fixed digit format)
WHILE %0<10
  DEFINE %0 %0+=1
LOOP
```

```
// Decrement Defined Variable %0 (10) by 1 until 00 (10 - 00 retains 2 fixed digit format)
WHILE %0>00
  DEFINE %0 %0-=01
LOOP
```

Notes: Nested WHILE & IF support. BREAK aborts. CONTINUE resumes at top. Multiple AND | OR logic (cannot mix AND / OR tests). Variables used for cond can be a text string, a Dynamic or an Environment Variable, or any combination thereof.

Conditional Tests

Usage: IF [NOT] {cond} THEN ..., IF [NOT] {cond} ... [ELSEIF [NOT] {cond} ...] [ELSE ...] ENDIF, & WHILE [[NOT] cond] ... LOOP
Returns TRUE or FALSE condition based upon the results of the test.

Command Syntax	Comments / Examples
==	EQUAL: ASCII hex compare by character and for length. TRUE: IF 123==123 IF "ABC"=="ABC" IF NOT ZYX==XYZ FALSE: IF 321==123 IF "ABC"=="AB" IF NOT XYZ==XYZ
>	GREATER: ASCII hex compare by character and for length. TRUE: 2>1 1>02 "ABCD">"ABC" abc>ABC FALSE: -1>-2 -02>-1 "ABCD">"abc" XYZ>abc
<	LESS: ASCII hex compare by character and for length. TRUE: 1<2 02<1 "ABC"<"ABCD" XYZ<xyz FALSE: -2<-1 -1<-02 "abc"<"ABCD" abc<XYZ
>=	GREATER OR EQUAL: ASCII hex compare by character and for length. (GREATER > and EQUAL == examples apply).
<=	LESS OR EQUAL: ASCII hex compare by character and for length. (LESS < and EQUAL == examples apply).
ACTIVE_TASK [server/]{task}	Check if task is active (running) on local / remote Server. Notes: Wild cards supported. server only required for Remote Server checks. Excludes task issuing check (on local Server).
CONSOLE_LOCKED	Verifies the lock status of the actively displayed Server Console screen. Note: Locked screens cannot be changed without first being unlocked.

CONSOLE_SCREEN {"screen"}	Checks if the name of the actively displayed Server Console screen matches "screen" (case sensitive match up to length of "screen").
CURRENT_SCREEN {"screen"}	Checks if the name of the current screen against which the active task performs keyin / screen interrogation commands matches "screen" (case sensitive match up to length of "screen").
ERRORLEVEL	Reports the result (TRUE = Error / FALSE = Success) of the preceding task command line.
EXIST [server/][[drive:]vol:]path\}{spec}	Confirms existence of a directory / file matching {spec}. Supports wild cards. server only required for Remote Server checks.
FILE_IN_USE [server/][[vol:]path\]{spec}	Checks if any file matching the specification is in use (open) on the Server. Supports wild cards. server only required for Remote Server checks.
LOADED [server/]{module[.ext]}	Checks if any module matching the specification is loaded on the Server. Note: server only required for Remote Server checks. CLIB & CLIB.NLM / MPS14 & MPS14.PSM both match, if loaded.
LOGGED_IN {user}	Checks if any user matching the specification is logged in to the Server. Supports wild cards.
MOUNTED [server/]{vol}	Checks if a volume matching the specification is mounted on the Server. server only required for Remote Server checks.
SCAN_FILE {"string"} {filespec}	Scans any file matching the file specification for "string". Notes: "string" must be enclosed in double quotes (""). Comparison is case sensitive. Supports wild cards in the file specification.
SCAN_SCREEN {"string"}	Scans the current screen against which the active task performs keyin / screen interrogation commands for "string". Notes: "string" must be enclosed in double quotes (""). Comparison is case sensitive. Only supports character (non-GUI) screens. Use CONSOLE SCREEN / CURRENT SCREEN to change the current task I/O screen.
SCAN_STRING {"substring"} {"string"}	Scans "string" for "substring". Note: Both strings must be enclosed in double quotes (""). Comparison is case sensitive.
SCREEN_LOCKED	Note: Obsolete (supported for backwards compatibility). Replacement: CONSOLE_LOCKED

Dynamic Variables

Dynamic Variables are used to store data in a reusable manner and can be used as a specification in any task command line.

Command Syntax	Comments / Examples
%alias_name%	<p>A name assigned to a previously Aliased Variable or a Defined Variable to replace the name previously used to reference the variable. (Refer to VARALIAS for more information on and examples of Aliased Variables.)</p> <p>Naming rules: Must begin and end with a percent sign (%); First character after initial percent sign (%) cannot be numeric (0-9); Maximum of 15 characters, including percent signs (%); Alias names cannot contain spaces.</p> <p>Note: When used within a task, an Aliased Variable is handled by the batch parser exactly the same as its corresponding Defined Variable.</p>
%0 - %9	Data variables specific to each running task in which

%VAR00% - %VAR99%	<p>alphanumeric string values can be stored and modified.</p> <p>Note: %0 - %9 are internally reserved names which correspond to the first 10 data variables (named %VAR00% - %VAR09% respectively, by default). %VAR00% - %VAR99% are default names for all 100 data variables available to each task which can be renamed (or aliased) using VARALIAS. When used within a task script, the batch parser will automatically replace any properly referenced variable name with any value stored in the associated variable. Upon initial task execution, %0 - %9 are assigned any arguments (options) which were specified on the TMRUN command line. Some of the Batch Commands which can be used to assign or modify the contents of Aliased Variables or Defined Variables include: CALC, DEFINE, PARSE, READ, REFORMAT, REPLACE, TOLOWER and TOUPPER.</p>
--------------------------	--

Environment Variables

Environment Variables contain real-time information about the Server which the batch parser automatically substitutes during processing.

Command Syntax	Comments / Examples
%AM_PM%	Substitutes 'am' or 'pm' (based on local 12 hour clock).
%CACHE_LRU_NUMERIC%	Substitutes the current Cache Least Recently Used (LRU) time as a string representing the days, hours, minutes and seconds (000:00:00:00 to 999:23:59:59 - 12 byte fixed length, zero filled).
%CACHE_LRU_SECS%	Substitutes the current Cache Least Recently Used (LRU) time as a numeric value representing the number of seconds (00000000 to 99999999 - 8 digit fixed length).
%CACHE_LRU_STRING%	Substitutes the current Cache Least Recently Used (LRU) time as a string representing the days, hours, minutes and seconds (range '1 secs' to '999 days 23 hrs 59 mins 59 secs' - variable length).
%CACHE_RATIO%	Substitutes the current Cache Buffer ration or percent of Server RAM available for File Cache Buffers (000 to 100 - 3 digit fixed length).
%CONN_ADDRESS_{spec}%	<p>Substitutes the IP/IPX network address for the specified connection.</p> <pre>// Use a specific connection (12) ECHO Conn 12 netaddr: %CONN_ADDRESS_12%</pre> <pre>// Use numeric value defined in %0 WRITE Conn %0 netaddr: %CONN_ADDRESS_%0%</pre> <p>Note: {spec} can be a numeric string or a Dynamic Variable. If an INACTIVE connection, returns zero filled address.</p>
%CONN_ID_{spec}%	<p>Substitutes the logged in User's Object ID (same as used for the SYS:MAIL directory) for the specified connection.</p> <p>Notes: If NOT-LOGGED-IN or INACTIVE, returns a 0. (Refer to %CONN_ADDRESS_% for usage examples.)</p>
%CONN_NAME_{spec}%	<p>Substitutes the logged in User's Name for the specified connection.</p> <p>Notes: If not a User connection, returns NOT-LOGGED-IN or INACTIVE. (Refer to %CONN_ADDRESS_% for usage examples.)</p>
%CONNS_ACTIVE%	<p>Substitutes the current active (allocated) connection count (0000 to 9999 - 4 digit fixed length).</p> <p>Note: Includes Licensed, Authenticated, NOT-LOGGED-IN,</p>

	NLM, and remote attachment connections.
%CONNS_IN_USE%	Substitutes the current active (allocated) connections in use count (0000 to 9999 - 4 digit fixed length). Note: Includes Licensed, Authenticated, NLM and remote attachment (excludes NOT-LOGGED-IN) connections.
%CONNS_MAX%	Substitutes the maximum number of active connections supported (0000 to 9999 - 4 digit fixed length). Note: NetWare v4/v5/v6 returns %CONNS_PEAK% + 1 since the control is by license, not connection, count.
%CONNS_PEAK%	Substitutes the peak number of active connections in use at any time since the Server was last booted (0000 to 9999 - 4 digit fixed length). Note: Includes Licensed, Authenticated, NOT-LOGGED-IN, NLM, and remote attachment connections.
%CONSOLE_SCREEN%	Substitutes the name of the current screen actively displayed on the Server Console.
%CPU_UTIL%	Substitutes the current Server CPU utilization percentage (000 to 100 - 3 digit fixed length).
%CURRENT_SCREEN%	Substitutes the name of the current screen against which the active task performs keyin / screen interrogation commands.
%CWD%	Substitutes the Current Working Directory (CWD) for the active task. Note: To change the CWD, use the CD / CHDIR Console Command.
%CX%	Substitutes the Directory Services Context for the Server (null string returned for NW v3 and Bindery only Servers).
%DAY%	Substitutes the current day of the month (01 to 31 - 2 digit fixed length).
%DAY_OF_WEEK%	Substitutes the current day of the week (Sunday - Saturday).
%DIR_FILE_[{spec}]%	Substitutes the name of the first/next file matching the initial specification (required for first pass/use). <pre>// Find first file in Current Working Directory (CWD) matching *.* DEFINE %1 %DIR_FILE_*. *% // Find next file matching last specified pattern (*.*) DEFINE %1 %DIR_FILE_% // Find first file matching SYS:\PUBLIC*. * DEFINE %1 %DIR_FILE_SYS:\PUBLIC*. *% // Find next file matching last specified pattern (SYS:\PUBLIC*. *) DEFINE %1 %DIR_FILE_% // Scan Current Working Directory (CWD) WHILE // If first loop (%1 == NULL - unused), use *.* to initiate scan & no spec (NULL) to find next IF "%1"==" " DEFINE %1 %DIR_FILE_*. *% ELSE DEFINE %1 %DIR_FILE_% ENDIF // If %DIR_FILE_% returns a NULL value (%1==" " - i.e., no more) then exit WHILE/LOOP (BREAK) IF "%1"==" " THEN BREAK // ECHO (display) the returned name</pre>

	<pre>ECHO %1 LOOP</pre> <p>Notes: A valid DOS search pattern must be specified in the initial (first) request. Subsequent (next) requests should not include a search pattern (i.e., specifying a pattern initiates a new search). A null string is returned if the search is exhausted or fails. If the initial (first) specification does not include a volume and/or path, the Current Working Directory (CWD) is used. A Dynamic Variable may be used for initial (first) specification, if it is defined as a valid DOS search pattern. Wild cards supported.</p>
<pre>%DIR_SUB_{spec}%</pre>	<p>Substitutes the name of the first/next directory matching the initial specification (required for first pass/use). Note: Refer to %DIR_FILE_% for usage examples.</p>
<pre>%DIR_TREE_{spec}%</pre>	<p>Substitutes the name of the first/next subdirectory in the specified directory (required for first pass/use) and changes the Current Working Directory (CWD) to correspond to the returned directory, traversing the directory tree below the initial (first) specification.</p> <pre>// Scan & display a specified directory tree WHILE // If first loop (%1 == NULL - unused), use SYS:PUBLIC to // initiate scan & no spec (NULL) to find next IF "%1"==" " DEFINE %1 %DIR_TREE_SYS:PUBLIC% ELSE DEFINE %1 %DIR_TREE_% ENDF // If %DIR_TREE_% returns a NULL value (%1==" " - i.e., // no more) then exit WHILE/LOOP (BREAK) IF "%1"==" " THEN BREAK // ECHO (display) the next level traversed in the CWD ECHO %CWD% LOOP</pre> <p>Notes: A valid DOS directory must be specified in the initial (first) request. Subsequent (next) requests should not include a specification (i.e., specifying a directory initiates a new search). A null string is returned if the search is exhausted or fails. The initial (first) specification becomes the base (or root) directory for traversing the tree. A Dynamic Variable may be used for initial (first) specification, if it is defined as a valid DOS directory. Unlike %DIR_SUB_% which only returns a matching directory name, the returned directory also becomes the CWD.</p>
<pre>%DOS_DRIVE_FREE_{?}%</pre>	<p>Substitutes the amount of free space (MBytes) on the specified or default (boot) DOS drive (0000 to 99999999 - 4 to 8 digits variable length).</p> <pre>// Store information for Drive C: in %1 DEFINE %1 %DOS_DRIVE_FREE_C%</pre> <pre>// Store information for the default (boot) drive in %1 DEFINE %1 %DOS_DRIVE_FREE_%</pre> <p>Note: If a DOS drive (A-Z) is not specified for ? then the default (boot) DOS drive information is returned.</p>
<pre>%DOS_DRIVE_SIZE_{?}%</pre>	<p>Substitutes the drive size (MBytes) for the specified or default (boot) DOS drive (0000 to 99999999 - 4 to 8 digits variable length).</p>

	Note: Refer to %DOS_DRIVE_FREE_% for usage examples.
%DOS_DRIVE_USED_[?]%	Substitutes the space used (MBytes) on the specified or default (boot) DOS drive (0000 to 99999999 - 4 to 8 digits variable length). Note: Refer to %DOS_DRIVE_FREE_% for usage examples.
%ELAPSED_HOURS%	Substitutes the number of hours that the current task has been running (00 to 4294967295 - 2 digits minimum length).
%ELAPSED_MINS%	Substitutes the number of minutes that the current task has been running (00 to 4294967295 - 2 digits minimum length).
%ELAPSED_SECS%	Substitutes the number of seconds that the current task has been running (00 to 4294967295 - 2 digits minimum length).
%ELAPSED_TIME%	Substitutes a text string representing the hours, minutes and seconds that the current task has been running (00:00:00 to 99999:59:59 - minimum length of 8 bytes).
%FILE_ACCESS_{spec}%	Substitutes the date {spec} was last accessed (yyyy/mm/dd - 10 bytes fixed length). // Store last access for LOGIN.DAT (located in CWD) in %1 DEFINE %1 %FILE_ACCESS_LOGIN.DAT% // Store last access for SYS:\PUBLIC\LOGIN.DAT in %1 DEFINE %1 %FILE_ACCESS_SYS:\PUBLIC\LOGIN.DAT% // Store SYS:\PUBLIC\LOGIN.DAT in %0 & last access for %0 in %1. DEFINE %0 SYS:\PUBLIC\LOGIN.DAT DEFINE %1 %DIR_FILE_%0% // Store file search pattern (*.*) in %0. // Store the directory to search (SYS:\PUBLIC) in %1. // Use %DIR_FILE_% to search for a match for the combined path/pattern (SYS:\PUBLIC*.*). // If a match is returned (%2 is not NULL), store the last access for %1\%2 in %3. DEFINE %0 *.* DEFINE %1 SYS:\PUBLIC DEFINE %2 %DIR_FILE_%1\%0% IF NOT "%2"==" " THEN DEFINE %3 %FILE_ACCESS_%1\%2% Notes: {spec} can be fully qualified or relative to the Current Working Directory (CWD) and must reference a valid directory or file. If {spec} does not exist, a null string is returned. If a Dynamic Variable is used for {spec}, it must be defined as a valid directory or file.
%FILE_ATTRIB_{spec}%	Substitutes the NetWare Attributes assigned to {spec}. Notes: For information on the returned attributes and their order, refer to the CHMOD / FLAG Extended Console Commands. Refer to %FILE_ACCESS_% for usage examples.
%FILE_CREATE_{spec}%	Substitutes the date {spec} was created (yyyy/mm/dd - 10 bytes fixed length). Note: Refer to %FILE_ACCESS_% for usage examples.
%FILE_MODIFIER_{spec}%	Substitutes the Object name that last modified {spec}. Note: Refer to %FILE_ACCESS_% for usage examples.
%FILE_NAME_DOS_{spec}%	Substitutes the full DOS Name Space path (VOL:\PATH\SPECNAME.EXT) for {spec}. Note: Refer to %FILE_ACCESS_% for usage examples.
%FILE_NAME_LONG_{spec}%	Substitutes the full Long Name Space path (Vol:\Path\SpecName.ext) for {spec}.

	Note: Refer to %FILE_ACCESS_% for usage examples.
%FILE_NAME_MAC_{spec}%	Substitutes the full Macintosh Name Space path (Vol:Path:SpecName.ext) for {spec}. Note: Refer to %FILE_ACCESS_% for usage examples.
%FILE_NAME_NFS_{spec}%	Substitutes the full NFS Name Space path (Vol:/Path/SpecName.ext) for {spec}. Note: Refer to %FILE_ACCESS_% for usage examples.
%FILE_OWNER_{spec}%	Substitutes the Object name assigned as Owner of {spec}. Note: Refer to %FILE_ACCESS_% for usage examples.
%FILE_SIZE_{spec}%	Substitutes the size of {spec} (0000000000 to 4294967295 - 10 digits fixed). Notes: Subdirectories return a size of 0. Refer to %FILE_ACCESS_% for usage examples.
%FILE_UPDATE_{spec}%	Substitutes the date {spec} was updated (modified) last (yyyy/mm/dd - 10 bytes fixed length). Note: Refer to %FILE_ACCESS_% for usage examples.
%HOUR%	Substitutes the current 12 hour format (01 to 12 - 2 digit fixed length)
%HOUR24%	Substitutes the current 24 hour format (00 to 23 - 2 digit fixed length)
%LICENSES_IN_USE%	Substitutes the current number of licenses in use (0000 to 9999 - 4 digit fixed length).
%LICENSES_MAX%	Substitutes the maximum number of licenses supported (0000 to 9999 - 4 digit fixed length).
%LICENSES_PEAK%	Substitutes the peak number of licenses in use since the Server was booted (0000 to 9999 - 4 digit fixed length).
%MINUTE%	Substitutes the current minute (00 to 59 - 2 digit fixed length).
%MONTH%	Substitutes the current month number (01 to 12 - 2 digit fixed length)
%MONTH_NAME%	Substitutes the current month name (January to December).
%NDAY_OF_WEEK%	Substitutes a number corresponding to the current week day (1 [Sunday] to 7 [Saturday] - 1 digit fixed length).
%NDAY_OF_YEAR%	Substitutes the current Julian day number in the year (001 to 365 [366 in Leap Years] - 3 digit fixed length).
%NW_SUBVERSION%	Substitutes the minor version of the NetWare OS in use (00 to 99 - 2 digit fixed length).
%NW_SUPPORTPACK%	Substitutes the NetWare Support Pack level (returns SP#, # is the level). Note: NetWare v5+ only (NetWare v3/v4 returns a null string).
%NW_VERSION%	Substitutes the major version of the NetWare OS in use (0 to 9 - 1 digit fixed length).
%PATH%	Substitutes the path, excluding the volume name, of the Current Working Directory (CWD). Note: SYS:\ -> <null> (no path) SYS:\SYSTEM -> \SYSTEM
%RECV_BUFFS_IN_USE%	Substitutes the number of Packet Receive Buffers (PRBs) allocated by NetWare (0 to 4294967295 - variable length).
%RECV_BUFFS_MAX%	Substitutes the maximum number of Packet Receive Buffers (PRBs) which can be allocated by NetWare (0 to 4294967295 - variable length).
%SCREEN_NAME%	Note: Obsolete (supported for backwards compatibility). Replacement: %CONSOLE_SCREEN%
%SECOND%	Substitutes the current second (00 to 59 - 2 digits fixed length).
%SERVER%	Substitutes the name of the Server.

%SERVER_NET%	Substitutes the Server IPX Internal Network address, if defined (hex value in ASCII format - 8 byte fixed length).
%TASK%	Substitutes the file name (filename.ext) for the active Task.
%TASK_EXT%	Substitutes the extension (no path or file name) for the active Task.
%TASK_FILE%	Substitutes the file name (no path or extension) for the active Task.
%TASK_LINE_NUM%	Substitutes the line number where located within the active Task.
%TASK_NAME%	Substitutes the full name (vol:\path\filename.ext) for the active Task.
%TASK_PATH%	Substitutes the path (vol:[\path]) for the active Task.
%TASK_SCREEN%	Substitutes the name of the screen created to receive the default I/O, as well as error messages, for the active task.
%TEMP_NAME%	Substitutes a unique temporary NetWare file name (in the format of _TM?????.TMP, where the ?'s are replaced with an ASCII representation of a unique decimal number; e.g., _TM01234.TMP).
%TM_REVISION%	Substitutes the revision level of the TaskMaster module in use (a to z or null - 1 byte max length).
%TM_SUBVERSION%	Substitutes the minor version of the TaskMaster module in use (01 to 99 - 2 digits fixed length).
%TM_VERSION%	Substitutes the major version of the TaskMaster module in use.
%TREE%	Substitutes the Directory Services Tree name in which the Server is located (null string returned for NW v3 and Bindery only Servers).
%UPTIME_NUMERIC%	Substitutes the current Server Up Time as a numeric string representing the days, hours, minutes and seconds (000:00:00:00 to 999:23:59:59 - fixed length, zero filled).
%UPTIME_STRING%	Substitutes the current current Server Up Time as a text string representing the days, hours, minutes and seconds ('1 secs' to '999 days 23 hrs 59 mins 59 secs' - variable length).
%VOL_FREE%	Substitutes the percentage of available space on the volume associated with the Current Working Directory (CWD), including space which is purgeable (000 to 100 - 3 digits fixed length).
%VOL_NAME%	Substitutes the volume name (VOL:.) associated with the Current Working Directory (CWD).
%VOL_PURGEABLE%	Substitutes the percentage of purgeable space on the volume associated with the Current Working Directory (CWD) (000 to 100 - 3 digits fixed length).
%VOL_SIZE% %VOL_SIZE_MB%	Substitutes the size (MBytes) of the volume associated with the Current Working Directory (CWD) (00000000 to 99999999 - 8 digits fixed length).
%VOL_USED%	Substitutes the percentage of used space on the volume associated with the Current Working Directory (CWD) (000 to 100 - 3 digits fixed length).
%YEAR%	Substitutes the current year (1900 to 2100 - 4 digits fixed length).

TaskMaster / TMLite Quick Reference

Extended Console Commands

(As of: v4.18 / Updated: 15 January 2008)

Executable via TMConsole (Shell) prompt or within a Batch / Task script

Syntax : { } required [] optional | 'or' **Options** : Must be separated by space
Column > : Output redirection supported (i.e., for logging - must be last item on command line)
 (overwrite: >vol:path\name.ext append: >>vol:path\name.ext)

Local Server Commands

(Supported by TaskMaster & TMLite *unless* noted otherwise.)

Command Syntax	Options	>	Comments
ABORT {cmd task}	/# # (numeric) instance /A All matching instances		Cancel the specified {cmd} or {task}. The /# and /A options apply when aborting specific instances of multiple active SYNC commands or tasks.
APPEND [[d: vol:]path\]{src} [[d: vol:]path\]{dest}		X	Append {src} file specification to {dest} file specification.
CD [[vol:]path] CHDIR [[vol:]path]			Change Current Working Directory (CWD) to path. Note: If no specification provided show CWD.
CHMOD [[vol:]path\]{spec} [[+ -]attr]	(Refer to FLAG)	X	(Refer to FLAG)
CHOWN [[vol:]path\]{spec} [user]	/DO Dirs only /FO Files only /H Hidden/System /S Subdir recursion	X	List/Set owner for dir & files matching {spec}. Notes: List mode if no [user] specified. {spec} must be a valid file specification or search pattern.
CLEAR CONNS {#} [# # # ...]		X	Clear (kill) specified connections.
CLEAR USERS {user} [user user user ...]		X	Clear (kill) connection with logged in Object names matching {user}. Note: Supports wild cards.
COPY [[d: vol:]path\]{src} [[d: vol:]path\]{dest}	/D Decompress dest (NW Co) /I Ignore errors /O Owner set (def) /-O not /T Trustee set (def) /-T not	X	Basic local Server file copy (NSS / TFS Volumes & DOS partition). Notes: Attributes, date/time stamps, IRM owner, and trustee rights are set to match the source, by default Supports wild cards.
DEL [[d: vol:]path\]{spec} DELETE [[d: vol:]path\]{spec}	/Y Confirm delete all		Delete any file(s) which match {spec}. Note: Supports wild cards.
DELTREE [[vol:]path\]{spec}	/# Repeat process up to /# times		Recursive deletion of

	/I Ignore non-critical errors /K Keep dirs (files only) /P Purge deleted dirs/files /V Verbose listing /Y Confirm delete all		dirs & files which match {spec}. Note: Supports wild cards.
DIR [[d: vol:]path\][spec]	/A Attr info /A=search /B Bare list /D Dir list /-D no dirs /F File list /-F no files /H Hidden/System /L= Len (size) search /N NS info /N=search /O Owner info /O=search /-O Bad /Q Quiet mode (summary list only) /R Rights info /R=search /-R Bad /S Subdir recursion /T Time info /T=search /W Wide list /X DOS list (no NS info)	X	List dirs & files matching [spec] and optional search criteria. Notes: Lists all, if no [spec]. Supports wild cards.
DUMP [[d: vol:]path\]{file}			Hexadecimal list of {file} contents.
ERASE [[d: vol:]path\]{spec}			(Refer to DEL or DELETE)
FILE CLOSE [[vol:]path\]{spec}	/S Subdir recursion	X	Close file(s) matching {spec} without clearing (killing) the associated connection(s). Notes: File(s) are internally flushed prior to close. Supports wild cards.
FILE UNLOCK [[vol:]path\]{spec}	/S Subdir recursion	X	Unlock file(s) matching {spec} without clearing (killing) the associated connection(s). Notes: Only Server-based locks. Supports wild cards.
FIND "string" [[vol:]path\]{spec}	/C Count lines /I Case insensitive /L List files only /N Line Numbers /S Subdir recursion /X DOS name	X	List file(s) matching {spec} which contain "string". Notes: /C overrides /I and /N. Supports wild cards.
FLAG [[vol:]path\]{spec} [[+ -]attr]	/DO Dirs only /FO Files only /H Hidden/System /S Subdir recursion	X	Set attributes for dirs &/or files matching {spec}. Note: Lists current attributes if no [attr] specified.
FLAGDIR [vol:]{spec} [[+ -]attr]	/H Hidden/System /S Subdir recursion	X	Set attributes for dirs matching {spec}. Note: Lists current attributes if no [attr] specified.
GRANT {rights} [[vol:]path\]{spec} {user group}		X	Set specified {rights} to {user} or {group} for Dir(s)/File(s)

		matching {spec}. Note: Supports wild cards.
LIST CONNS {#} [# # # ...]	/A Address /O Object type	X List info about the specified connection (s).
LIST MODULES [nlm]	/I Info (resources)	X List info on any modules matching [nlm]. Notes: Lists all loaded modules if no [nlm]. Supports wild cards.
LIST SCREENS		X List NLM screens & which NLM created each.
LIST SESSIONS	/I Info (resources)	X List any active TMRremote sessions. Note: * <i>Not supported by TMLite.</i> *
LIST USERS [user]	/A Address /O Object type	X List any connections with logged in Object names matching [user]. Notes: Lists all logged in conns if no [user]. Supports wild cards.
LIST VOLUMES [vol]		X List info on any mounted volume matching [vol]. Note: List all volumes if no [vol].
MD [[d: vol:]path\]{dir} MKDIR [[d: vol:]path\]{dir}	/D Create as DOS Name Space /L Create as Long Name Space /M Create as Mac Name Space /N Create as NFS Name Space	Create {dir}. Note: Supports DOS partition.
PURGE [[vol:]path\]{spec}	/# deleted age (min) /A All (subdirs) /D Dirs processed /Q Query (list only) /Z Zero length files	X Purge deleted files matching {spec} in [[vol:]path]. Note: Purge all files in CWD if no {spec}.
RD [[d: vol:]path\]{dir} RMDIR [[d: vol:]path\]{dir}		Remove {dir}. Note: Supports DOS partition.
REN [[d: vol:]path\]{spec} [path\]{spec} RENAME [[d: vol:]path\]{spec} [path\]{spec}	/C Change owning Name Space /D DOS Name Space /L Long Name Space /M Mac Name Space /N NFS Name Space	Rename files. Notes: Same drive / volume only. Support wild cards. Supports DOS partition. The /C option requires a Name Space option and changes the owning Name Space. If a Name Space is specified without the /C option, only the specified Name Space name is modified.
RENDIR [[d: vol:]path\]{spec} [path\]{spec}	/C Change owning Name Space /D DOS Name Space /L Long Name Space	Rename directory. Notes: Same drive / volume only. The /C

	/M Mac Name Space /N NFS Name Space		option requires a Name Space option and changes the owning Name Space. If a Name Space is specified without the /C option, only the specified Name Space name is modified.
REVOKE {rights} [[vol:]path\]{spec} {user group}		X	Revoke {rights} from {user} or {group} for dirs & files matching {spec}. Note: Supports wild cards.
SORT [[vol:]path\]{src} [[vol:]path\]{dest} {keys}	/D Descending sort /I Insensitive case /N Null records excl. /P Pack data for keys	X	Sort {src} into {dest} using {keys}.
SC [cmdline] SYSTEM CONSOLE [cmdline]			Change the active screen on the Server Console to the System Console screen. Note: Optional [cmdline] can be passed to System Console prompt.
TASKLIST		X	List active tasks.
TMCMD {cmd} [...]			Launches {cmd} (Local Console Command) as a single command task (i.e., batch processed).
TMCONFIG ADD OPERATOR {obj_name[.cx]} TMCONFIG DEL OPERATOR {obj_name[.cx]} TMCONFIG LIST OPERATOR [spec] TMCONFIG ADD USER {obj_name[.cx]} TMCONFIG DEL USER {obj_name[.cx]} TMCONFIG LIST USER [spec] TMCONFIG LOG CLEAR TMCONFIG LOG SAVE [[vol:]path\]filename.ext TMCONFIG TMREMOTE {user_class} {rights} TMCONFIG TMREMOTE LIST TMCONFIG TMREMOTE PASSWORD			Manage the TaskMaster configuration, including: Add/Del/List TM Operator(s); Add/Del/List TM Users(s); Clear/Save Master log file; Define/List TM Remote Console Users/Rights/Password Note: * <i>Only the LOCAL options are supported by TMLite.</i> *
TMHELP [cmd]			On-line help for [cmd] Note: General help if no [cmd] specified.
TMINFO		X	Registration information.
TMRELOAD			Reload Server module (NLM).
TMRESET			Causes the NLM to re-read the configuration

		& schedule files.
TMRUN [[vol:]path\]{task} ["arg"] ["arg" "arg" ...]		Launch processing of {task}.
TMSCHEDULE [task] TMSCHEDULE ADD {task} {date} {time} TMSCHEDULE DEL {task} [/# /A]	/# matching task number # /A all matching tasks	X Add/Del/List task schedule. ADD: {date} = date (mm/dd), every month (dd) or weekday (Y N for SMTWTFS). ADD: {time} = time (hh:mm), every hour (:mm) or interval (mm). Notes: Use DOS 8.3 name only for task, no path or Long name. Wild cards supported for [task], except for ADD/DEL.
TMSERVER	CX - list DS context DS - list full DS Server name TREE - list DS tree UPTIME - list Server up-time	X List Server information.
TMSMTP {svc.domain.sub} [[vol:]path\]{filename.ext} TMSMTP {###.###.###.###} [[vol:]path\]{filename.ext}		Connection to the specified SMTP Server (first specification, named or dot notated), then send the specified file (second specification) as a self contained E-Mail message (i.e., TO: / FROM: contained therein). Note: * <i>Not supported by TMLite.</i> *
TMUNLOAD		Unloads the NLM.
TREE [[vol:]path]	/A ASCII indent character /B Bare list /D Delimited data /F Fixed length fields /I Dir/File info /S Size not formatted /X DOS name only	X List directory tree for [path]. Note: List Current Working Directory (CWD) tree, if no [path] specified.
TYPE [[d:]vol:]path\]{file}		List the contents of the specified {file}. (ASCII dump.)
USERLIST [user]	(Refer to LIST USERS)	X (Refer to LIST USERS)
VOLINFO [vol]		X (Refer to LIST VOLUMES)
WHEREIS {spec}	/D incl Dirs (def) /-D excl /F incl Files (def) /-F excl /I Info	X List dirs / files matching {spec} on Current Working Volume (CWV).
WHOHAS {spec}	/S Subdir recursion	X List connection(s) / module(s) with any file open matching {spec}
XCOPY [[d:]vol:]path\){src} [[d:]vol:]path\){dest}	/A Archive copy	X Enhanced local Server

/D Decompress dest (NW Co) /E Empty Dir copy /G Gridlock (error if dest open) /H Hidden/System /I Ignore errors /M (/A & reset attr) /N Nice (shared RO) /-N Excl mode /O Owner set (def) /-O not /R overwrite RO /S Subdir recursion /T Trustee set (def) /-T not	copy (NSS / TFS Volumes & DOS partition). Notes: Attributes, date/time stamps, IRM owner, and trustee rights are set to match the source, by default Supports wild cards. Supports DOS partition.
--	--

Remote Server Commands

(Supported by TaskMaster, not supported by TMLite.)

Command Syntax	Options	> Comments
SCOPY [[vol:]path\]{lsrc} rserver/vol:path\[rdest] SCOPY rserver/vol:path\{rsrc} [[vol:]path\[ldest]	/B Block update (def) /-B full copy /C Compress xfer /-C Raw /D Decompress dest (NW Co) /I Ignore errors /O Owner set (def) /-O not /T Trustee set (def) /-T not /+IP[=?] if IP avail, use ? pkt size /+NCP[=?] use NCP & ? pkt size /# Summary processing statistics	X Basic file copy between Servers. Block Update: If /-B is not specified, Block (Delta Change) Update logic is used to minimize the amount of data transferred for file updates. Compression: If neither /C nor /-C specified (default), intelligent compression used (compression on attempted on uncompressed file types). Gridlock: If destination file is Open (InUse), it is closed (default) unless /G option is specified which causes an InUse error. IP Support: If both Servers are NetWare v5+ with IP and a stable comm link can be established, data transfer will utilize IP (UDP) by default. While the default maximum calculated packet size cannot be increased, it can be decreased via /+IP=x option (x = desired packet size - range: 1024 - 32768). Using /+NCP or /+NCP=x option (x = desired packet size - range: 1024 - 32768) disables IP (UDP) support and forces use of NCP Extensions. Notes: Attributes, date/time stamps, IRM owner, and trustee

			rights are set to match the source, by default Wild cards are supported for Local-to-Remote operations only.
STASKLIST {rserver}		X	List active tasks on {rserver}.
STMINFO {rserver}		X	Registration info for {rserver}.
SXCOPY [[vol:]path\]{lsrc} rserver/vol:path\[ldest] SXCOPY rserver/vol:path\{rsrc} [[vol:]path\[rdest]	/A Archive copy /B Block update (def) /-B full copy /C Compress xfer /-C Raw /D Decompress dest (NW Co) /E Empty Dir copy /G Gridlock (error if dest open) /H Hidden/System /I Ignore errors /J Jumble (encrypt) data /M (/A & reset attr) /N Nice (shared RO) /-N Excl mode /O Owner set (def) /-O not /R overwrite RO /S Subdir recursion /T Trustee set (def) /-T not /X eXclude non-DOS criteria /+IP[=?] if IP avail, use ? pkt size /+NCP[=?] use NCP & ? pkt size /# Summary processing statistics	X	Enhanced file copy between Servers. Block Update: If /-B is not specified, Block (Delta Change) Update logic is used to minimize the amount of data transferred for file updates. Compression: If neither /C nor /-C specified (default), intelligent compression is used (compression attempted on uncompressed file types). Gridlock: If destination file is Open (InUse), it is closed (default) unless /G option is specified which causes an InUse error. IP Support: If both Servers are NetWare v5+ with IP and a stable comm link can be established, data transfer will utilize IP (UDP) by default. While the default maximum calculated packet size cannot be increased, it can be decreased via /+IP=x option (x = desired packet size - range: 1024 - 32768). Using /+NCP or /+NCP=x option (x = desired packet size - range: 1024 - 32768) disables IP (UDP) support and forces use of NCP Extensions. Notes: Attributes, date/time stamps, IRM owner, and trustee rights are set to match the source, by default Wild cards and the options /A /E /M /S are supported for Local-to-

		Remote operations only.
TMSCMD {rserver} {cmd}		Submit {cmd} to {rserver}.
TMUPDATE [rserver] [rserver rserver ...]	/U Update / Upgrade only	X Distributes/reloads TaskMaster NLMs to specified [rserver] Remote Servers. Notes: Updates know Servers running TaskMaster NLM if r [rserver] specified. Wild cards supported.

SYNC: Replicate Files / Synchronize Data

(Supported by TaskMaster, not supported by TMLite.)

Command Syntax	Options	> Comments
SYNC [[vol:]path\]{src} vol:path [<excl.lst] SYNC [[vol:]path\]{src} remote/vol:path [<excl.lst] SYNC LIST	/A Add (if not on dst) /B Block update (def) /-B full copy /C Compress xfer /-C Raw /D Del (if not of src) /E Excl src access /-E Shared R/O /F File info compare /G Gridlock (error if dest open) /H Hidden/System /I Ignore errors /J Jumble (encrypt) data /K Keep dirs (with /D) /L Log command line /M Mod (src newer dest) /O Owner set (def) /-O not /Q Query mode /R Replace (copy all) /S Subdir recursion /T Trustee set (def) /-T not /U Upd (dst newer src) /V Ver (src/dst mismatch) /W Warnings/Errors only /X eXclude non-DOS processing /+ENT=# max entries per directory /+EXC=# max entries for exclusion /+IP[=?] if IP avail, use ? pkt size /+MIG=[Y N] process migrated files /+NCP[=?] use NCP & ? pkt size LIST List active SYNC processing	X Replicate files &/or synchronize data between Local directories and directories on Local o Remote Servers based upon the options specified. Block Update: If /-B i not specified, Block (Delta Change) Updat logic is used to minimize the amount of data transferred for file updates. Compression: If neither /C nor /-C specified (default), intelligent compressio used (compression on attempted on uncompressed file types). Gridlock: If destinatio file is Open (InUse), i is closed (default) unless /G option is specified which cause an InUse error. IP Support: If both Servers are NetWare v5+ with IP and a stable comm link can be established, data transfer will utilize IP (UDP) by default. While the default maximum calculated packet size cannot be increased, it can be decreased via /+IP=x option (x = desired packet size - range: 1024 - 32768). Using /+NCP

		<p>or /+NCP=x option (x = desired packet size range: 1024 - 32768) disables IP (UDP) support and forces use of NCP Extensions. For more detailed information, refer to the: SYNC FAQ</p> <p>Notes: Attributes, date/time stamps, IRM owner, and trustee rights are set to match the source, by default. Specific source dirs, files & file patterns can be excluded using a redirection list ([<excl.lst]).</p>
--	--	--