



DECUS makes it work

<i>MMMM</i>	<i>MMMM</i>	<i>CCCCCCCC</i>	<i>EEEEEEEEEE</i>
<i>MM MM</i>	<i>MM MM</i>	<i>CCC CCC</i>	<i>EE EEE</i>
<i>MM MM MM</i>	<i>MM</i>	<i>CC CC</i>	<i>EE</i>
<i>MM MMM</i>	<i>MM</i>	<i>CC</i>	<i>EEEEEEE</i>
<i>MM M</i>	<i>MM</i>	<i>CC</i>	<i>EEEEEEE</i>
<i>MM</i>	<i>MM</i>	<i>CC CC</i>	<i>EE</i>
<i>MM</i>	<i>MM</i>	<i>CCC CCC</i>	<i>EE EEE</i>
<i>MM</i>	<i>MM</i>	<i>CCCCCCCC</i>	<i>EEEEEEEEEE</i>

COMMANDLINE EDITOR

This manual describes MCE V4.28
from September 1998

originally developed by

D. Sponza (YU) Oct-81

Since 29-Jul-86 Supported by

Hans Hamakers
pAkUiT International
Elsbes 40 , 3069 LN Rotterdam
The Netherlands

<http://www.hamakers.nl>

Different “evolutions” from DECUS tapes (sometimes called CLE) where
collected and merged and new features where added

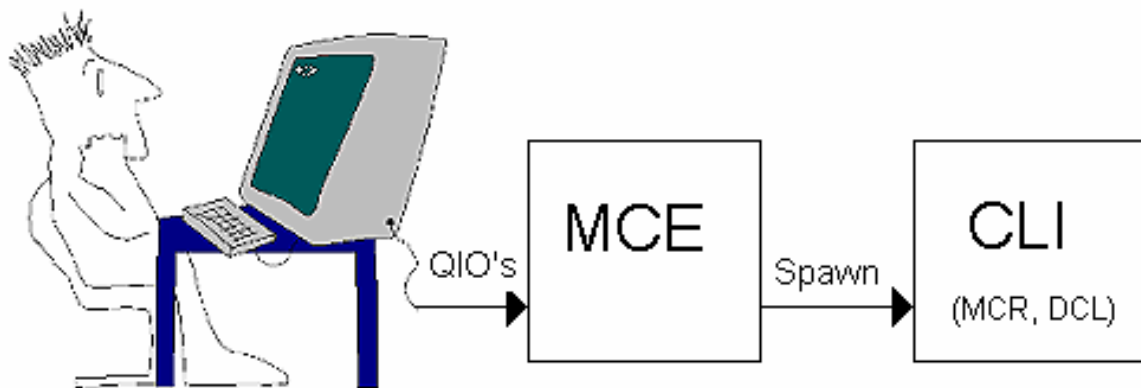
Table of Contents

1. INTRODUCTION	4
2. OPTIONS	5
2.1 MULTIPLE CLI SUPPORT	5
2.2 INIT-FILE SUPPORT	5
2.3 COMPOUND COMMANDLINE SUPPORT	6
2.4 STATUSLINE SUPPORT	7
2.5 STARTUP AND EXIT MESSAGES	7
2.6 REMOTE TERMINAL (RT: AND HT:) DETECTION.....	8
2.7 VTPLUS SUPPORT	8
2.8 VT4PLUS SUPPORT	8
2.9 TANDBERG TDV2230 FUNCTION KEY SUPPORT.....	9
2.10 EDT-KEYPAD EDITING SUPPORT.....	9
2.11 USER PROMPT SUPPORT.....	9
2.12 EXTENDED PROMPT SUPPORT	9
2.13 TERMINAL PASSWORD LOCKING.....	9
2.14 AUTOMATIC RETURN TO INSERT OR OVERSTRIKE MODE.....	9
2.15 TERMINAL TIME-OUT SUPPORT	9
3. STARTUP AND EXIT	11
3.1 COMMANDLINE EDITOR STARTUP.....	11
3.2 COMMANDLINE EDITOR EXIT	11
4. COMMANDLINES	12
5. COMMAND SYNONYMS	13
6. FUNCTIONKEYS	14
7. INTERNAL COMMANDS (MCE XXXX)	15
7.1 MCE CHAIN [FILE]	15
7.2 MCE CLEAR	15
7.3 MCE CMSZ N.....	16
7.4 MCE ECHO ON/OFF.....	16
7.5 MCE EXIT	16
7.6 MCE FISZ N.....	16
7.7 MCE FREE	16
7.8 MCE INSERT ON/OFF	17
7.9 MCE KEYPAD ON/OFF	17
7.10 MCE LIST	17
7.11 MCE LOCK	17
7.12 MCE OVERSTRIKE ON/OFF.....	17
7.13 MCE PROMPT ON/OFF	17
7.14 MCE PURGE	18
7.15 MCE READ [FILE]	18
7.16 MCE REPLACE [FILE]	18
7.17 MCE RING	18
7.18 MCE STATUS [ON/OFF].....	18
7.19 MCE SVINTERNALS ON/OFF.....	18
7.20 MCE SVOLD ON/OFF.....	18
7.21 MCE TIMEOUT N/ON/OFF	18
7.22 MCE UPFIND ON/OFF	19
7.23 MCE USPROMPT [USERPROMPT].....	19

7.24 MCE VERSION	19
7.25 SOME NOTES ON INIT-FILES.....	19
8. RECALL FUNCTION	20
9. HELP FILE	21
10. LINE EDITING COMMANDS	22
10.1 NON-KEYPAD EDITING AND CURSOR MOVEMENT COMMANDS	22
10.2 EDT-KEYPAD EDITING AND CURSORMOVEMENT COMMANDS	23
10.2.1 VT2+ Keyboard Lay-out.....	24
10.2.2 VT1xx Keypad Lay-out.....	25
10.2.3 VT52 Keypad Lay-out	26
10.2.4 Keypad editing and cursormovement keys.....	27
11. BUILDING MCE.....	29
12. KNOWN PROBLEMS.....	31
12.1 DCL AND INSTALLED TASKS	31
12.2 MCE ON RT:, HT: AND LAT TERMINALS	31
12.3 REMARKS ON 8-BIT TERMINALS AND LINES	31
APPENDIX A.....	33
EXAMPLE OF AN INIT-FILE	33
APPENDIX B.....	34
SAMPLE MCE BUILD-SESSION (RSX-11M WITH MCR AND NO SAVED ANSWERS).....	34
APPENDIX C.....	39
SAMPLE MCE BUILD-SESSION (RSX-11M WITH MCR AND SAVED ANSWERS)	39

1. Introduction

MCE is an RSX-task through which you can communicate with the CLI (CommandLine Interpreter, e.g. MCR or DCL) currently active at your terminal.



MCE let you edit your Commandlines in an EDT-like way in keypad or non-keypad mode. MCE stores the Commandlines for you in a FIFO-buffer for later retrieval and you can define your own commands in the same way as in OpenVMS.

2. Options

A list of optional features follows below. They can individually be selected at build time.

MCE V4.28 - Commandline Editor options :

- Multiple CLI support
- Init-file support
- Compound Commandline support
- Statusline support
- Startup and exit messages
- Remote terminal (RT: and HT:) detection
- VT2plus support *
- VT4plus support †
- TDV2230 function key support
- EDT-Keypad editing support
- User Prompt support
- Extended Prompt support
- Terminal password locking
- Automatic return to Insert or Overstrike mode
- Terminal Time-Out support

2.1 Multiple CLI support

When multiple CLI support is included MCE uses the prompt from the CLI that is active at the invoking terminal. Furthermore different Init-files for MCR, DCL or other CLI's can exist.

2.2 Init-file support

When Init-file support is included MCE reads initialization files which can contain Command synonym definitions, Internal- and CLI-commands. Files are searched in this order:

When multiple CLI support is included :

```
LB: [1,2]MCEINI.xxx, followed by  
SY: [CurDir]MCEINI.xxx or SYS$LOGIN:MCEINI.xxx
```

* In this manual VT2plus means support for VT2xx, VT3xx, VT4xx and VT5xx terminals.

† In this manual VT4plus means support for VT4xx, and VT5xx terminals.

When multiple CLI support is not included :

```
LB: [1,2]MCEINI.CMD, followed by  
SY: [CurDir]MCEINI.CMD or SYS$LOGIN:MCEINI.CMD
```

where “xxx” is the name of the CLI defined for the terminal invoking MCE and “CurDir” is the default directory at the startup of MCE.

This algorithm divides command definition into two groups: A system-wide set in LB: [1,2] followed by users private set in SY: [CurDir] or SYS\$LOGIN:

MCE can be configured to always read the user Init-file from SYS\$LOGIN: on systems that support Logical Names (Only possible on M-Plus). This makes you independent from where you start MCE.

Overriding the default filename in the Commandline which invokes MCE offers the possibility to tailor an application specific command set.

Comments are possible behind “!” which is the comment character for CLI’s.

An Example of an Init-file can be found in Appendix A.

2.3 Compound Commandline support

MCE allows definitions of compound commands. Using the ampersand “&” sign an MCE command can be defined as a sequence of multiple CLI commands :

```
e.g.:  +>STAT := TIM & PIP /LI & ACT /ALL ..... etc.
```

This defines STAT as a sequence of CLI-commands that are executed one after another. MCE waits for completion of one command before activating the next.

Also it can be done directly:

```
e.g.:  +>MAC @TASKASM & TKB @TASKTKB
```

The “&” **MUST** be preceded by at least one space. This is done to let commands like “PIP /TD&/LI” still work.

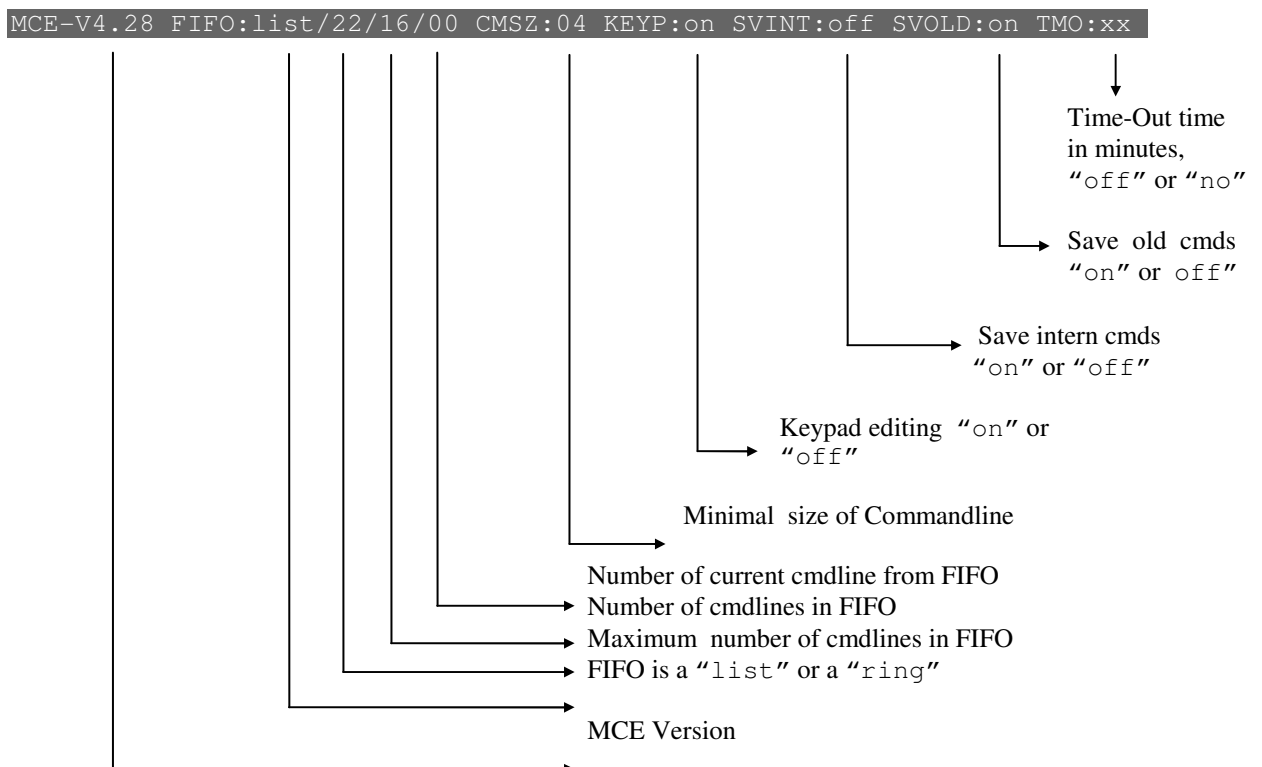
When a command in a compound Commandline gives a non-successful exitstatus, MCE terminates the compound Commandline there, showing the exitstatus and the command that caused it.

A compound Commandline may also contain Internal Commands (Commands starting with “MCE”) in mixture with normal CLI commands. Only the first command is checked for translation.

2.4 Statusline support

Line 24 on an ANSI CRT terminal can be used for status information. The Statusline shows the different settings and FIFO-buffer parameters of MCE and can be switched on or off with **MCE STATus ON/OFF**. For non ANSI terminals like VT52's only a "SNAPSHOT" is possible with **MCE STATus**. The Statusline is then displayed only once.

The line displays the following :



2.5 Startup and exit messages

The startup and exit messages like :

```
MCE -- CLI Commandline Editor V4.28
and
MCE -- Exit
```

are optional. They can be selected at build time.

MCE identifies its self with :

```
MCE -- MCR Commandline Editor V4.28
```

when no multiple CLI support was selected at build time.

2.6 Remote terminal (RT: and HT:) detection

When MCE is started via a remote terminal problems can occur if the local system is VMS with its Commandline editor enabled. The way MCE acts when it detects if it is started via a remote terminal can be specified at build time. One of the following options can be selected :

- Give the following message:

```
MCE -- ** WARNING ** Started on a Remote Rerminal.  
      If your local system is OpenVMS with its Commandline  
      editor enabled please type "MCE EXIT"
```

- Give the following message and exit :

```
MCE -- Started on a Remote Terminal. Exiting .....
```

- Do nothing.

2.7 VTPlus support

VT2plus support enables the use of VT2+[‡] keyboards.

The keys <F11>, <F12> and <F13> work as <ESC>, <BS> and <LF>. The keys <HELP>, <DO> etc. can also be used and it is possible to define the Functionkeys <F6> .. <F20> like :

```
F6 := TIM
```

Pushing <F6> then gives the time of day. (See Functionkeys).

2.8 VT4plus support

On a VT4+[§] terminal it is possible to use <F1> .. <F5> as normal Functionkeys. MCE enables you to define those keys in the same way as other Functionkeys when this support is included with MCEBLD.

[‡] VT2+ in this manual stands for VT2xx and later models.

[§] VT4+ in this manual stands for VT4xx and later models.

2.9 Tandberg TDV2230 function key support

The keys of a Tandberg TDV2230 terminal can be defined like the ones on a VT2+. (See Functionkeys).

2.10 EDT-Keypad editing support

Some edit functions can be performed on a VT52/VT1xx/VT2+-keypad, similar to K52, KED and EDT. The Keypad-editing can be switched on or off with **MCE KEYPad ON/OFF**.

2.11 User Prompt support

The MCE prompt can be specified by the user with **MCE USPRompt UserPrompt**. A User Prompt can contain up to 20 characters.

The return to the default prompt type **MCE USPRompt**.

2.12 Extended Prompt support

The MCE prompt can show if Overstrike or Insert mode is active with "+" or "-". Also it can display if keypad editing is enabled when the Statusline is off with ":".

e.g.: +:>

The Extended Prompt can be switched on or off with **MCE PROMpt ON/OFF**.

2.13 Terminal password locking

When selected this option enables the **MCE LOCK** command. It prompts for a password and verification and locks the terminal. When the password is entered again the terminal is ready to use again.

This is all like the LOCK command from Digital's Terminalservers.

2.14 Automatic return to Insert or Overstrike mode

Default Overstrike or Insert mode stays active until it is changed by <CTRL/A> or <F14>. MCE can be build so that it sets one of the modes active by default when it prompts for a new command. This can be changed with the **MCE INSErt ON/OFF** and **MCE OVERstrike ON/OFF** commands.

2.15 Terminal Time-Out support

When terminal Time-Out support is included and the terminal is not used within a given time the terminal is logged-out. (at TT0: MCE only exits).

Optionally the time-out value can be changed with ***MCE TIMOut nnn*** to be 2 to 999 minutes or can be enabled or disabled with the ***MCE TIMOut ON/OFF*** command. The ***MCE TIMOut*** command **MUST** be included during build time to let this work. (So system managers, do not be afraid, if YOU do not want it, they will not get it).

3. Startup and Exit

3.1 Commandline Editor Startup

Invoke MCE with command "`MCE`", preferably in your LOGIN file by "`.XQT MCE`". If Init-file support is included a file of Command-Synonyms, CLI-Commands and/or Internal-Commands will be read and executed at MCE startup. The file may be specified in the form "`MCE <startup-file>`". If no startup file is specified in the Commandline (the normal case), the default Init-files are searched for. (See Init-files).

3.2 Commandline Editor Exit

To exit MCE use the ***MCE EXIT*** internal command or `<CTRL/Y>`.

4. Commandlines

When MCE is ready to accept a Commandline it displays a prompt. How the prompt looks depends on the chosen options and setup of MCE. (See Extended Prompt). The maximum length of Commandlines is 78 characters.

If a Commandline starts with "MCE", it is interpreted as an Internal Command. Other Commandlines are checked for translation and the result is spawned to the current CLI.

Special action is performed with "BYE" and "LOG[out]". "BYE" and when in DCL "LOG[out]" result in a spawn of "BYE" to MCR and MCE exits immediately. Note that you may define any other string (e.g. "GOODBYE") to result in "BYE" or "LOGOUT".

All Commandlines can be preceded by and contain one or more TAB's and/or spaces.

Normally all Commandlines equal to or longer than a given length are saved in a FIFO-buffer. This length can be changed with the **MCE CMSZ n** command.

The maximum number of Commandlines the FIFO-buffer can hold can be specified with the **MCE FISZ n** command. When the maximum number of Commandlines is present, or when no buffer pool space is available to hold more entries, the new Commandline is saved and the oldest disappears.

Retrieved Commandlines from the FIFO-buffer are only saved again if they are edited or when the saving of "old" Commandlines is enabled by **MCE SVOLD ON** and this Commandline is not the newest in the FIFO. This prevents filling up your FIFO with one command by continuously repeating the last command.

Internal commands are saved when this is enabled by **MCE SVInternals ON**. (Save "INTERNAL" commands).

5. Command synonyms

A Command synonym can be defined by:

```
+>Syn := <Command1> [ & <Command2> [ & ..... ] ].
```

where "Syn" is the command synonym which will be substituted by the text at the right side of "=="

```
e.g.:  +>HE := HELP PIP
       +>HE                               Will request "HELP PIP"
```

Note : The "==" must be preceded and followed by at least one TAB or space.

With Keypad-mode "OFF" the keystrokes <PF1> , <BLUE> or <ESC>P translate the command synonym without execution.

If you enter a command synonym followed by additional text, this additional text will normally be appended as a whole.

```
e.g.:  +>HE /RE                               Will request "HELP PIP /RE"
or      +>HE /RE<PF1>                         Will be translated and displayed
result: +>HELP PIP /RE
```

A command synonym can be deleted by redefining it to null.

```
e.g.:  +>HE :=                               Delete above definition
```

Abbreviate commands by separating the optional part by "*"

```
e.g.:  +>HP*IP := HELP PIP
```

This command can now be requested with:

```
+>HP or HPI or HPIP etc.
```

More sophisticated is the Parameter Substitution, similar to the indirect commandfile processor.

```
e.g.:  +>DIR*ECTORY      := PIP 'P1'/LI
       +>COP*Y           := PIP 'P2'/NV/CD='P1'

       +>DIR              Will be translated into "PIP /LI"
       +>DIREC FIL        Will be translated into "PIP FIL/LI"
       +>COPY A.CMD [1,2] Will simulate the DCL COPY command
```

Note that command synonyms may be performed from Init-files.

6. Functionkeys

You can request a command with Functionkeys <PF1>, <PF2>, <PF3> or <PF4>.

The keys <BLUE>, <RED> and <GREY> on a VT52 are synonym to <PF1>, <PF2> and <PF3>.

It is also possible to request commands with control keys (e.g. <CTRL/G>).

With a VT2+ keyboard you can also define the Functionkeys <F6>..<<F20> and <FIND> <INSERT HERE> etc. (Only when VT2plus support is included).

With a VT4+ <F1>..<<F5> can be defined when this support is included.

The keys of a Tandberg TDV2230 keyboard can be defined like the ones on a VT2+. (Only when TDV2230 support is included).

Define the command translation as "<key> := command".

e.g.:	+>PF2 := TIM	<i>Redefine <PF2> or <RED> key</i>
	+><PF2>	<i>to display the current time</i>
	+>F6 := ACT	<i>Define F6 function key</i>
	+><F6>	<i>to display the active tasks</i>

The keys <FIND> etc. on a VT2+ keyboard must be entered as F\$a where "a" is the first character of the key function. (English keynames).

e.g.: F\$P for <PREV-SCREEN>

Define a control key as follows :

e.g.:	+>^G := SET /UIC	<i>Define <CTRL/G> key</i>
	+><CTRL/G>	<i>to display the current UIC</i>

Note that the following control-keys CAN NOT be defined:

```
<CTRL/M> ( = <RETURN> )
<CTRL/O>
<CTRL/Q>
<CTRL/S>
<CTRL/[> ( = <ESC> )
```

Next to the rest of the Alphabetic characters the following keys can be defined :

```
<CTRL/\>
<CTRL/]>
<CTRL/^>
<CTRL/?>
```

Note : A Userdefined key overrules the function that the key may already have for MCE.

7. Internal Commands (MCE xxxx)

If a Commandline starts with “MCE” an Internal Command is assumed. Following the verb **MCE** a four character action specifier defines the action wanted.

e.g.: **MCE CLEA** this command clears the FIFO-buffer.

Here follows a list of all internal commands :

- **MCE CHAIn [file]**
- **MCE CLEAR**
- **MCE CMSZ n**
- **MCE ECHO ON/OFF**
- **MCE EXIT**
- **MCE FISZ n**
- **MCE FREE**
- **MCE INSErt ON/OFF**
- **MCE KEYPad ON/OFF**
- **MCE LIST**
- **MCE LOCK**
- **MCE OVERstrike ON/OFF**
- **MCE PROMpt ON/OFF**
- **MCE PURGe**
- **MCE READ [file]**
- **MCE REPLace [file]**
- **MCE RING**
- **MCE STATusline [ON/OFF]**
- **MCE SVINternals ON/OFF**
- **MCE SVOLd ON/OFF**
- **MCE TIMOut n|ON/OFF**
- **MCE UPFind ON/OFF**
- **MCE USPRompt [UserPrompt]**
- **MCE VERSion**

7.1 MCE CHAIn [file]

Chain from one Init-file to another, that is start or continue reading commands from the specified file. This command is a synonym for **MCE READ [file]**. This command is only available when Init-file support is included in MCE.

7.2 MCE CLEAR

This command deletes all commands currently in the FIFO-buffer.

7.3 MCE CMSZ *n*

Changes minimal command size to "*n*". Only Commandlines that are equal to or longer than "*n*" characters are stored in the FIFO-buffer. Set the length to 79, if for any reason, you do not want to save commands. The default setting can be specified at MCE build time.

7.4 MCE ECHO ON/OFF

When "ON", MCE echoes Commandlines to TI:. This is useful in Init-files to display messages etc. or to see the translation of command synonyms. The default setting is "OFF". (See example in Appendix A).

7.5 MCE EXIT

The task MCE exits.

7.6 MCE FISZ *n*

Changes the maximum number of Commandlines which can be saved in the Commandline FIFO-buffer. Note that the actual number of Commandlines that can be saved depends on the size of the Commandlines and the available MCE poolspace.

Information about MCE's poolspace can be obtained by the **MCE FREE** command. More poolspace can be created by installing MCE with a larger increment. "*INS \$MCE/INC=xxx*".

A value of 2400, which is the default value that is used during taskbuilding/linking, will be enough in most cases. If "*n*" is smaller than the active number of Commandlines stored, MCE deletes the oldest Commandlines from the FIFO-buffer until "*n*" is reached. The default setting can be specified at MCE build time.

7.7 MCE FREE

Displays MCE's Pool information like :

```
MCE -- Pool:xxx:yyy:zz.
```

```
xxx   = Size of largest block in bytes
yyy   = Total number of free bytes
zz    = Number of fragments
```

7.8 MCE INSErt ON/OFF

When "ON", MCE goes back to Insert mode when prompting for a new command. The default setting can be specified at MCE build time.

MCE INSErt ON Switches Auto-Overstrike mode off.
MCE OVERstrike ON Switches Auto-Insert mode off.

7.9 MCE KEYPad ON/OFF

Set Keypad editing on or off. (See EDT-Keypad editing). This command is only available when EDT-Keypad support is included in MCE.

7.10 MCE LIST

Makes command FIFO-buffer a list : <UP> and <DOWN> stop at the top and bottom of the FIFO. This is the default setting. (See also **MCE RING** command).

7.11 MCE LOCK

The LOCK command prompts for a Password and verification and locks the terminal until the password is typed again. Like the LOCK command from Digital's terminal servers. The maximum password-length is 8 characters.

This command is only available when Terminal Password locking support is included in MCE.

7.12 MCE OVERstrike ON/OFF

When "ON", MCE goes back to Overstrike mode when prompting for a new command. The default setting can be specified at MCE build time.

MCE OVERstrike ON Switches Auto-Insert mode off.
MCE INSErt ON Switches Auto-Overstrike mode off.

7.13 MCE PROMpt ON/OFF

Switches Extended Prompt on or off. The default setting can be specified at MCE build time. This command is only available when Extended Prompt support is included in MCE.

7.14 MCE PURGe

This command deletes all command synonyms currently in the synonym buffer.

7.15 MCE READ [file]

Read Commandlines from specified file. This command is only available when Init-file support is included in MCE.

7.16 MCE REPLace [file]

Combines the **MCE PURGe** and **MCE READ [file]** commands. This command is only available when Init-file support is included in MCE.

7.17 MCE RING

Makes command FIFO-buffer a ring : <UP> and <DOWN> roll through the buffer. The default setting is "LIST". (See also **MCE LIST** command).

7.18 MCE STATus [ON/OFF]

This command sets the Statusline on or off.. When no option is specified the Statusline is displayed only once as a snapshot. (See Statusline support)
The default setting can be specified at MCE build time. This command is only available when Statusline support is included in MCE.

7.19 MCE SVInternals ON/OFF

When "ON", Internal commands (Commandlines starting with "MCE") are saved in the FIFO-buffer, When "OFF" these commands are not saved. The default setting can be specified at MCE build time.

7.20 MCE SVOLd ON/OFF

When "ON", an "old" command (retrieved from the FIFO-buffer) is saved again in the FIFO-buffer if it is not the newest in the FIFO. When "OFF", "old" commands are not saved unless they are edited. The default setting can be specified at MCE build time.

7.21 MCE TIMOut n/ON/OFF

This command allows you to change the time-out value in minute that you specified during MCEBLD run. The value “n” may be anything in the range of 2 to 999. You may also enable or disable time-out with the **ON/OFF** parameter.

This command is only available when Settable Terminal Time-Out support is included in MCE.

7.22 MCE UPFind ON/OFF

Since Version 4.27 of MCE <UP> does a <RECALL> function (like <FIND>) when there is something on the commandline and the cursor is at the end of the line. This feature is included because many PC commandline editors have that feature. This is the default. The user can change to the old situation with the **MCE UPFind OFF** command.

7.23 MCE USPRompt [UserPrompt]

When “UserPrompt” is specified MCE uses this argument as the new prompt. It can contain up to 20 characters. When “UserPrompt” is not specified MCE uses the default prompt again. This command is only available when User Prompt support is included in MCE.

7.24 MCE VERSion

Shows the MCE Version identification.

7.25 Some notes on Init-files

When with the **CHAIN**, **READ** and **REPLace** commands no file is specified, the same filename convention is used as at startup: the files according to the current CLI are read, i.e. the command **MCE REPLace** results in a MCE restart, but the FIFO-buffer is kept. This is useful after a “SET TERMINAL <new-cli>” command. When a “space” is given as filename, only SY:[CurDir]MCEINI.xxx or SYS\$LOGIN:MCEINI.xxx is read. (See Init-file support).

8. Recall function

Recalling a command is done by entering the first part of a command which was previously executed, terminated by a RECALL keystroke. This will recall the last command already executed starting with the string entered. The command can be edited or executed by pressing the return key. A second RECALL keystroke searches further back in the FIFO-buffer. When no string is entered, the last defined string is taken.

Recall keystrokes are :

<FIND>
<ESC>Q

When anything is on the commandline	: <UP>
In non EDT-Keypad mode	: <PF2> or <RED>
In EDT-Keypad mode	: <PF3> or <GREY>

Example:

Previously entered Commandlines :

```
+>DIR MCE*. *  
+>MAC @MCEASM  
+>TKB @MCETKB  
+>DMP X.DAT  
+>TIM
```

```
+>D<RECALL>
```

Will recall the last Commandline starting with "D" :

```
+>DMP X.DAT
```

Another <RECALL> keystroke will recall an older command from the FIFO starting with "D"

```
+>DIR MCE*. *
```

```
+>DI<RECALL>
```

Will recall :

```
+>DIR MCE*. *
```

Note : Since Version 4.27 of MCE <UP> does a <RECALL> function when there is something on the commandline and the cursor is at the end of the line. This feature is included because many PC commandline editors have that feature. This is the default. The user can change to the old situation with the **MCE UPFind OFF** command.

9. Help file

A helpfile named MCE.HLP customized to the selected options is created during MCE build time. The helpfile can be created in two different versions.

VERSION 1 :

You can access this version by typing "HELP MCE [topic.....]" but you have to include a reference like :

```
1 MCE
@MCE
```

in MCR.HLP and DCL.HLP

VERSION 2 :

You can access this version by typing "HELP /MCE [topic.....]".

This version does not need any editing of MCR.HLP and DCL.HLP

Note that if you do not build the helpfile together with MCE, you can not use the <NEXT-SCREEN> and <PREV-SCREEN> keys to walk through the helpfile.

10. Line editing commands

For line-editing there are two modes :

- Non-keypad mode
- Keypad mode

In the non-keypad mode you control MCE with Control-keys or Escape-sequences. In the keypad mode you can use the keypad of a VT52, VT1xx or a VT2+ terminal to give commands to MCE.

The advantage of this mode is that you can edit the current Commandline in the same way as you edit a line in K52, KED or EDT.

10.1 Non-keypad editing and cursor movement commands

Cursor movements	
Move Cursor Left	<LEFT> <CTRL/D>
Move Cursor Right	<RIGHT> <CTRL/F>
Move Cursor to Begin Of Line	<BACKSPACE> <CTRL/H> <F12>**
Move Cursor to End Of Line	<CTRL/E>
Move one Word to the right or move from EOL to BOL	<TAB> <CTRL/I>

Delete commands	
Delete Character Left of Cursor	<DELETE>
Delete Character at Cursor	<CTRL/V>
Delete Word Left of Cursor	<LINEFEED> <CTRL/J> <F13>**
Delete Word Right at Cursor	<CTRL/W>
Delete from Cursor to BOL	<CTRL/U>
Delete from Cursor to EOL	<CTRL/K>
Delete whole line	<CTRL/C>

Control commands	
Rewrite Line	<CTRL/R>
Switch between Overstrike and Insert mode	<CTRL/A> <F14>**

** On a VT2+ terminal only when VT2plus support is included

Save and execute	
Execute Command	<RETURN> <ENTER>
Execute Command without waiting	<ESC><ESC> <DO> **
Execute Cmd and leave FIFO-pointer where it is	<CTRL/X>
Save Command without executing	<CTRL/N>
Enable output from other tasks. Detach for 10 sec.	<CTRL/T>
Enable output from other tasks. Stop; Resume with "UNST MCE"	<CTRL/P>

Non-keypad Command buffer manipulation keys	
Display contents of FIFO-buffer	<PF4> <ESC>S
Retrieve previous Command	<UP> <CTRL/B>
Reverse of <UP> (next command)	<DOWN>
Recall Command from FIFO	<PF2> <RED> <FIND> ** <ESC>Q <UP> ††

Non-keypad Help keys	
Gives HELP about MCE	<HELP> ** <CTRL/?>
Gives Next HELP screen	<NEXT-SCR> **
Gives Previous HELP screen	<PREV-SCR> **

Non-keypad Command translation keys	
Translate Command synonym without execution	<PF1> <BLUE> <ESC>P
Show Command Synonym Buffer	<PF3> <GREY> <ESC>R

10.2 EDT-keypad editing and cursormovement commands

While in keypad-mode the non-keypad commands are valid, except those commands that are redefined.

†† See "MCE UPFind" command

10.2.1 VT2+ Keyboard Lay-out

F11	F12	F13	F14	HELP	DO
(ESC)	(BS) BEGLIN	(LF) DELWRD RIGHT	INS/OVR	HELP	EXECUTE NOWAIT

			PF1	PF2	PF3	PF4
(FIND) RECALL	INSERT HERE	REMOVE	GOLD	KPHELP	RECALL	DELLIN UNDLIN
SELECT	PREV HELP (SCREEN)	NEXT HELP (SCREEN)	TRANSL	SHOSYN VERS	SHOFIF FREE	DELWRD UNDWRD
	PREV COMM.		DETACH SUSPND		CUT PASTE	DELCHR UNDCHR
<==	NEXT COMM.	==>	WORD	EOL DELLIN	DO STAY	ENTER
			BEGLIN CLRLIN		SELECT RESET	SAVE

10.2.2 VT1xx Keypad Lay-out

UP	DOWN	LEFT	RIGHT
PREV. COMM.	NEXT CPMM.	<==	==>

PF1	PF2	PF3	PF4
GOLD	KPHELP	RECALL	DELLIN UNDLIN
TRANSL	SHOSYN VERS	SHOFIF FREE	DELWRD UNDWRD
DETACH SUSPND		CUT PASTE	DELCHR UNDCHR
WORD	EOL DELLIN	DO STAY	ENTER
BEGLIN CLRLIN		SELECT RESET	SAVE

10.2.3 VT52 Keypad Lay-out

BLUE	RED	GREY	UP
GOLD	KPHELP	RECALL	PREV COMM.
TRANSL	SHOSYN VERS	SHOFIF FREE	NEXT COMM.
DETACH SUSPND		CUT PASTE	==>
WORD	EOL DELLIN	DO STAY	<==
BEGIN CLRLIN		SELECT RESET	ENTER SAVE

10.2.4 Keypad editing and cursormovement keys

Note : <KPx> means key x on the keypad.

Control commands		
Select lower function	<PF1> <BLUE>	GOLD

Cursormovements		
Move Cursor to Begin Of Line	<KP0>	BEGIN
Move Cursor to End Of Line	<KP2>	EOL
Move one Word to the right or move from EOL to BOL	<KP1>	WORD

Delete commands		
Delete Character at Cursor	<KP, >	DELCHR
Delete Word Right at Cursor	<KP->	DELWRD
Delete from Cursor to EOL	<PF4> <BLUE><KP2> <PF1><KP2>	DELLIN
Delete Whole line	<PF1><KP0> <BLUE><KP0>	CLRLIN

Undelete commands		
Undelete Character	<PF1><KP, >	UNDCHR
Undelete Word	<PF1><KP->	UNDWRD
Undelete Line	<PF1><PF4>	UNDLIN

Cut and paste commands		
Set Begin Of Textmarker BOT	<SELECT> <KP.>	SELECT
Remove text from BOT to cursor and store it in the pastebuffer	<REMOVE> <KP6>	CUT
Insert text-string from pastebuffer starting at cursor	<INSERT> <BLUE><KP6> <PF1><KP6>	PASTE
Reset selected range	<PF1><KP.> <BLUE><KP.>	RESET

Save and execute		
Execute Command	<ENTER>	EXECUTE
Execute Cmd and leave FIFO-pointer where it is	<KP3>	DO-STAY
Save Command without executing	<PF1><ENTER> <BLUE><ENTER>	SAVE
Enable output from other tasks Detach for 10 sec.	<KP4>	DETACH
Enable output from other tasks Stop; Resume with "UNST MCE"	<PF1><KP4> <BLUE><KP4>	SUSPND

Command buffer manipulation keys		
Display contents of FIFO-buffer	<KP9>	SHOFIF
Recall Command from FIFO	<PF3> <GREY>	RECALL

Command translation keys		
Translate Command synonym without execution	<KP7>	TRANSL
Show Command Synonym Buffer	<KP8>	SHOSYN

Miscellaneous keys		
!Show version	<PF1><KP8> <BLUE><KP8>	VERS
Show free pool	<PF1><KP9> <BLUE><KP9>	FREE
Keypad Lay-out	<PF2> <RED>	KPHELP

11. Building MCE

This section deals with the generation of the MCE-task. The distribution contains the following files.

- MCE.MAC *Macro source file*
- MCEBLD.CMD *ICP commandfile to build MCE*
- MCEREL.TXT *Release notes*
- MCE.DOC *This manual in Microsoft Word 6.0 format*

To build The MCE-task you need the following files :

- MCE.MAC
- MCEBLD.CMD
- LB:[1,1]VMLIB.OLB

You start the building by typing :

`@MCEBLD` *(works with MCR and DCL)*

Next you are asked to answer questions about which options you want to be included in MCE. The questions that are asked are about :

- MULTIPLE CLI SUPPORT
- INIT-FILE SUPPORT
- COMPOUND COMMANDLINE SUPPORT
- STATUSLINE SUPPORT
- STARTUP AND EXIT MESSAGES
- TYPE OF RT: AND HT: DETECTION
- VT2plus SUPPORT
- VT4plus SUPPORT
- TANDBERG TDV2230 SUPPORT
- EDT-KEYPAD SUPPORT
- USER PROMPT SUPPORT
- EXTENDED PROMPT SUPPORT
- TERMINAL PASSWORD LOCKING
- AUTO INSERT/OVERSTRIKE DEFAULTS
- SAVE OLD COMMANDS DEFAULT
- SAVE INTERNAL COMMANDS DEFAULT
- TERMINAL TIME-OUT SUPPORT
- MINIMAL CMD-LINE LENGTH
- MAXIMUM ENTRIES IN FIFO
- HELP FILE

MCEBLD.CMD creates the following files:

- MCESAVED.DAT *Saved Answers*
- MCE.HLP *Help file*

- MCEPRE.MAC *Conditional assembly prefix file*
- MCEASM.CMD *MACRO commandfile*
- MCETKB.CMD *TKB/LINK commandfile*

Then MCEBLD.CMD invokes:

```
MAC @MCEASM
TKB @MCETKB or LINK @MCETKB
```

When present MCE will be built with FCSFSL.STB or FCSRES.STB

Sample build-sessions can be found in Appendices B and C

12. Known problems

12.1 DCL and installed tasks

MCE has some problems with DCL and installed tasks. DCL does not wait for installed tasks to complete. Compare the two examples below:

MCR

```
+>INS $FOO/TASK=FOO
+>RUN FOO
>
```

DCL

```
+$ INS $FOO/TASK=FOO
+$ RUN FOO
+$
```

To solve this add the following line to MCEINI.DCL :

```
R*UN := RUN 'P1'/STATUS:TASK
```

12.2 MCE on RT:, HT: and LAT terminals

A terminal on an RT:, HT: or LAT line that has a task running that attaches the terminal can give some problems. When the line becomes disconnected by network Time-Out's or manual interference the system tries to log-out the line.

On those lines BYE can not do I/O if the attached task has a priority higher than BYE. This is an RSX problem. MCE is a task that attaches the terminal and can give problems as mentioned. If BYE is installed with a higher priority than MCE (e.g. MCE 60 and BYE 70) the problem with MCE is solved.

12.3 Remarks on 8-Bit terminals and lines

A VT2+ terminal in 7-bit mode always sends "normal" Escape-sequences :

```
PF1 = ESC [ P          (33, 133, 120)
F20 = ESC ? 3 4 ~      (33, 77, 63, 64, 176)
```

The EBC, RPA, PASTHRU settings of the TI: do not have effect on the processing of those keys.

A VT2+ terminal in 8-bit mode sends other sequences :

```
PF1 = SS3 P          (217, 120)
F20 = CSI 3 4 ~      (233, 63, 64, 176)
```

In this case the EBC, RPA, PASTHRU settings of the TI: do have effect. Although they do different things one of the options MUST be set to "on" to let MCE see the keypad keys the right way.

The IO.RST used by MCE sees the following terminating characters in IOSTB+1 :

EBC	RPA	PASTHRU	SS3	CSI	MCE
-----	-----	---------	-----	-----	-----

0	0	0	0	33	not OK
0	0	1	17	33	OK
0	1	0	17	33	OK
0	1	1	17	33	OK
1	0	0	217	233	OK
1	0	1	217	233	OK
1	1	0	217	233	OK
1	1	1	217	233	OK

Note : Always look at the release notes in MCEREL.TXT to see the latest information.

APPENDIX A

- Example of an Init-file

```
!
!      LB:[1,2]MCEINI.MCR
!
MCE      STATUS      ON
MCE      SVINTERN    OFF
MCE      USPROMPT    Oldie>
!
?                := HELP
AS*SEMBLE        := MAC '@P1'ASM
CO*PY            := PIP 'P2'/NV='P1'
D*IR             := DIR
FO              := FOR '@P1'FTN
F20             := CLR
LI*NK           := TKB '@P1'TKB
MO*VE           := PIP 'P2'/NV='P1' & PIP 'P1';0/DE/NM
RE*NAME         := PIP 'P2'='P1'/RE
R*UN            := RUN
BLD             := MAC '@P1'ASM & TKB '@P1'TKB
!
! Next command let you change to DCL and read the MCEINI.DCL file(s)
!
      GODCL                := SET /DCL=TI: & MCE REPLACE
!
! MCR Commands .....
!
      CLR                  ! Clear screen
      MCE ECHO ON          ! Echo on
!
! ***WARNING*** -- The system will be down for PM next Friday
!
      MCE ECHO OFF          ! Echo off
      TIM                  ! Show time and date
!
! End of LB:[1,2]MCEINI.MCR
!
```

APPENDIX B

- Sample MCE build-session (RSX-11M with MCR and no saved answers)

```
>@mceblld
```

```

+-----+
|  +-----+  |
|  /         \  |
| / D E C U S \ |   D E C U S  makes it work
| \   P D P   /  |
|  \   S I G   /  |
|  +-----+  |
+-----+

```

MCEBLD.COMD -- Commandfile to build the MCE Commandline editor V4.28

This Commandfile builds YOUR version of MCE. It prompts for several options to be build in MCE. It creates a conditional assembly definition file called MCEPRE.MAC, two Commandfiles MCEASM.COMD and MCETKB.COMD and optionally a helpfile MCE.HLP. If you already have those files you can restart MCEBLD with "@MCEBLD MAC" to assemble and taskbuild or "@MCEBLD LINK" to taskbuild.

MCEBLD can use saved answers from a previous session.

MCEBLD can use the saved answers as default and still ask questions or it can directly use these answers and skip the questions which answers are defined.

MCEBLD -- Do you have a saved answer file [D:N] :

MCEBLD can save answers for a future session

MCEBLD -- Do you want to save answers [D:N] : y

MCEBLD -- Name of the saved answer file [D:MCESAVED.DAT] :

The standard CLI (Commandline Interpreter) of RSX is MCR. When your system supports the use of other CLIs such as DCL than include multiple CLI support in MCE.

MCEBLD -- Want the multiple CLI support [D:N] ? y

When Init-file support is included MCE reads initialization files which can contain Commanddefinitions Internal- and CLI-Commands.

Files are searched in this order:

LB:[1,2]MCEINI.xxx, followed by SY:[CurDir]MCEINI.xxx,

where "xxx" is the name of the CLI defined for the terminal invoking MCE and "CurDir" is the default directory at the startup of MCE.

Sample MCE build-session (RSX-11M with MCR and no saved answers)

This algorithm divides Commanddefinitions into 2 groups:
 A system-wide set in LB:[1,2], followed by a users private set.

MCEBLD -- Want the MCE Init-file support [D:N] ? y

MCE allows definitions of compound Commands.
 Using the ampersand (&) sign one can define an MCE Command as a
 sequence of multiple CLI Commands :

E.g. +>STAT := TIM & PIP /LI & ACT /ALL etc.

or directly: E.g. +>MAC @TASKASM & TKB @TASKTKB

MCEBLD -- Want the compound Command support [D:N] ? y

It is possible to include statusline support.
 The statusline shows the different settings and FIFO-parameters of
 MCE and can be switched on/off with MCE STATus on/off
 Line 24 is used for status information.

MCEBLD -- Want Statusline/Statusmessage support [D:N] ? y

MCEBLD -- Want statusline ON by default [D:N] ?

The startup and exit messages like :

"MCE -- CLI Commandline Editor V4.28" and "MCE -- Exit"

are optional.

MCEBLD -- Want Startup and exit messages [D:N] ? y

When starting MCE via a remote terminal (RT: or HT:) problems may occur
 if the local system is VMS with its Commandline editor enabled.
 MCE can detect if it was started via a remote terminal and can do
 3 different things :

1. Give the following message:

MCE -- ** WARNING ** Started on a Remote Terminal.
 If your local system is OpenVMS with its Commandline
 editor enabled please type "MCE EXIT"

2. Give the following message and exit :

MCE -- Started on a Remote Terminal. Exiting

3. Do nothing

MCEBLD -- Which option for RT: [D:1] : 2

MCEBLD -- Which option for HT: [D:1] : 2

VT2plus support enables the use of VT2xx, VT3xx, VT4xx and VT5xx
 terminals. The keys <F11>, <F12> and <F13> work as <ESC>, <BS> and
 <LF>. The keys <HELP>, <DO> etc. can also be used and it is possible

Sample MCE build-session (RSX-11M with MCR and no saved answers)

to define the Functionkeys <F6>..<<F20> like :

```
"F6 := TIM"
```

pushing <F6> gives the time of day then.

MCEBLD -- Want the VT2plus support [D:N] ? y

On a VT4xx and a VT5xx terminal it is possible to use <F1>..<<F5> as normal Functionkeys.

VT4plus support enables you to define those keys too if you answer "Y" to the following question.

MCEBLD -- Want the VT4plus support [D:N] ? y

The keys of a Tandberg TDV2230 terminal can also be defined.

MCEBLD -- Want the Tandberg TDV2230 function key support [D:N] ?

Some edit functions can be performed on the VTxxx keypad, similar to EDT, KED and K52.

The Keypad-editing can be switched on/off with MCE KEYPad on/off

MCEBLD -- Want the EDT-Keypad editing support [D:N] ? y

MCEBLD -- Want EDT-Keypad editing ON by default [D:N] ? y

The MCE prompt can show if overstrike or insert mode is active by "+" and "-".

It can also display if EDT-Keypad editing (when selected) is enabled when the statusline is off by ":"

e.g. "+:>"

The Extended Prompt can be switched on/off with MCE PROMpt on/off

MCEBLD -- Want the Extended Prompt [D:N] ? y

MCEBLD -- Want Extended Prompt ON by default [D:N] ? y

MCE can accept a User Prompt.

This prompt can be specified with MCE USPRompt <UserPrompt>.

The maximum prompt size is 20 characters.

MCEBLD -- Want the User Prompt [D:N] ? y

MCE can lock a terminal with a password.

Almost like the LOCK Command from Digital's terminal servers.

The maximum password-length is 8 characters.

MCEBLD -- Want the terminal password locking [D:N] ? y

Sample MCE build-session (RSX-11M with MCR and no saved answers)

By default overstrike or insert mode stays active until it is changed with <CTRL/A> or <F14>. MCE can be build so that it sets one of the modes active by default when it prompts for a new Command. This can be changed with the MCE INSErt on/off and the MCE OVERstrike on/off Commands.

MCEBLD -- Want MCE to set insert mode back by default [D:N] ? y

Default "old" Commands (retrieved from the FIFO and not edited) are saved again in the FIFO. MCE can be build so that it does not save them again.

This mode can be changed with the MCE SVOLd on/off Command.

MCEBLD -- Want MCE to save "Old" Commands by default [D:N] ? y

Default internal Commands are saved in the FIFO. MCE can be build so that it does not save them.

This mode can be changed with the MCE SVINtern on/off Command.

MCEBLD -- Want MCE to save Internal Commands by default [D:N] ? y

When terminal Time-Out support is included the terminal is logged out after a given Time-Out time (TT0: only exits MCE). Optionally you can select a fixed Time-Out time or select that the user can change the time.

MCEBLD -- Want the terminal Time-Out support [D:N] ? y

MCEBLD -- Want to enable the user to change the values [D:N] ? y

MCEBLD -- Enter default number of Time-Out minutes [2..999 D:4] : 360

MCEBLD -- Want the Time-Out to be ON by default [D:N] ? y

Only Commandlines with a length greater than or equal to a given length are saved in the FIFO.

The value can be changed with the MCE CMSZ n Command.

Use a value of 79 if you don't want to save at all.

MCEBLD -- Enter minimum Commandline length [1..79 D:4] : 1

Specify the default maximum number of Commands which can be saved in the FIFO. The value can be changed with the MCE FISZ n Command.

Note that the actual number of Commands that can be saved depends on the size of the Commands and the available MCE pool space.

Information about MCE's poolspace can be obtained by the MCE FREE Command.

More poolspace can be created by installing MCE with a larger increment. INS \$MCE/INC=xxx

A value of 2400 will be enough in most cases.

MCEBLD -- Enter maximum number of entries in FIFO [1..99 D:22] : 18

Sample MCE build-session (RSX-11M with MCR and no saved answers)

A helpfile MCE.HLP customized to your selections is created dynamically if you answer "Y" to the following question. Note that if you don't build the helpfile now, you can't use the <NEXT-SCREEN> and <PREV-SCREEN> keys to walk through the helpfile.

MCEBLD -- Create helpfile on LB: [D:N] ? y

MCEBLD -- What is your Help Directory [D:[1,2]] :

The helpfile can be created in two different versions.

VERSION 1 : You can access this one by typing HELP MCE [topic.....]
but you have to include a reference like :
1 MCE
@MCE
in MCR.HLP and DCL.HLP

VERSION 2 : You can access this one by typing HELP /MCE [topic.....]
This version does not need any editing of MCR.HLP and
DCL.HLP

MCEBLD -- Do you want VERSION 2 [D:N] ? y

MCEBLD -- When ready and successful copy MCE.TSK to LB: device [D:N] ? y

MCEBLD -- What is your Task Directory [D:[1,54]] :

MCEBLD -- Install task [D:N] ? y

MCEBLD -- Print listing and map [D:N] ?

MCEBLD -- All questions asked

MCEBLD -- Creating saved answer file MCESAVED.DAT

PIP MCESAVED.DAT/PU/NM

MCEBLD -- Creating helpfile LB:[1,2]MCE.HLP

PIP LB:[1,2]MCE.HLP/PU/NM

MCEBLD -- Creating configuration prefixfile MCEPRE.MAC

PIP MCEPRE.MAC/PU/NM

MCEBLD -- Creating MAC-Commandfile MCEASM.CMD

PIP MCEASM.CMD/PU/NM

MCEBLD -- Creating TKB-Commandfile MCETKB.CMD

PIP MCETKB.CMD/PU/NM

MCEBLD -- Assembling MCE

MAC @MCEASM

PIP MCE.OBJ/PU/NM,MCE.LST

MCEBLD -- Building MCE

TKB @MCETKB

PIP MCE.TSK/PU/NM,MCE.MAP

PIP LB:[1,54]/NV/FO/CO=MCE.TSK

ABO ...MCE

REM ...MCE

INS LB:[1,54]MCE

MCEBLD -- Procedure done

>

APPENDIX C

- Sample MCE build-session (RSX-11M with MCR and saved answers)

```
>@mceblld
```

```

+-----+
|   +-----+   |
|  /   \   \   |   D E C U S   makes it work
| / D E C U S \ |
| \   P D P   / |
|  \   S I G   / |
|   +-----+   |
+-----+

```

MCEBLD.COM -- Commandfile to build the MCE Commandline editor V4.28

This Commandfile builds YOUR version of MCE. It prompts for several options to be build in MCE. It creates a conditional assembly definition file called MCEPRE.MAC, two Commandfiles MCEASM.COM and MCETKB.COM and optionally a helpfile MCE.HLP. If you already have those files you can restart MCEBLD with "@MCEBLD MAC" to assemble and taskbuild or "@MCEBLD LINK" to taskbuild.

MCEBLD can use saved answers from a previous session.

MCEBLD can use the saved answers as default and still ask questions or it can directly use these answers and skip the questions which answers are defined.

```

MCEBLD -- Do you have a saved answer file [D:N] : Y
MCEBLD -- Name of the saved answer file [D:MCESAVED.DAT] : [1,2]MCESAVED.DAT
MCEBLD -- Do you want to skip the questions [D:N] : Y

```

MCEBLD can save answers for a future session

```

MCEBLD -- Do you want to save answers [D:N] : Y
MCEBLD -- Name of the saved answer file [D:[1,2]MCESAVED.DAT] :

```

```

*****
-- Multiple CLI support included      : T
*****
-- MCE Init-file support              : T
*****
-- Compound Command support          : T
*****
-- Statusline support                 : T
-- Statusline ON by default           : F
*****
-- Startup and Exit messages support: T
*****
-- RT messages                        : T
-- RT Exit                           : F
-- HT messages                        : T
-- HT Exit                           : F

```

Sample MCE build-session (RSX-11M with MCR and saved answers)

```

*****
-- VT2plus support : T
*****
-- VT4plus support : F
*****
-- Tandberg TDV2230 support : F
*****
-- EDT-Keypad support : T
-- EDT-Keypad ON by default : T
*****
-- Extended Prompt support : T
-- Extended Prompt ON by default : T
*****
-- User Prompt support : T
*****
-- Password Locking : T
*****
-- Insert mode default : T
-- Overstrike mode default : F
*****
-- Save OLD Commands : T
*****
-- Save Internal Commands : T
*****
-- Time-Out Support : T
-- Time-Out Support change : T
-- Time-Out value : 360
-- Time-Out default ON : T
*****
-- Minimum Commandline length : 1
*****
-- Maximum FIFO length : 18
*****

A helpfile MCE.HLP customized to your selections is created
dynamically if you answer "Y" to the following question.
Note that if you don't build the helpfile now, you can't use
the <NEXT-SCREEN> and <PREV-SCREEN> keys to walk through the helpfile.

MCEBLD -- Create helpfile on LB: [D:Y] ?
-- Help directory : [1,2]
*****
-- VERSION 2 Helpfile : T
*****

MCEBLD -- When ready and successful copy MCE.TSK to LB: device [D:Y] ?
MCEBLD -- What is your Task Directory [D:[1,54]] :
MCEBLD -- Install task [D:Y] ?
MCEBLD -- Print listing and map [D:N] ?

MCEBLD -- Use the sourcefile [7,136]mce.mac [D:N] ? Y
MCEBLD -- All questions asked .....

MCEBLD -- Creating saved answer file MCESAVED.DAT
PIP MCESAVED.DAT/PU/NM
MCEBLD -- Creating helpfile LB:[1,2]MCE.HLP .....
PIP LB:[1,2]MCE.HLP/PU/NM
MCEBLD -- Creating configuration prefixfile MCEPRE.MAC .....
PIP MCEPRE.MAC/PU/NM
MCEBLD -- Creating MAC-command file MCEASM.CMD .....
PIP MCEASM.CMD/PU/NM

```

```
MCEBLD -- Creating TKB-command file MCETKB.CMD .....
PIP MCETKB.CMD/PU/NM
MCEBLD -- Assembling MCE .....
MAC @MCEASM
PIP MCE.OBJ/PU/NM,MCE.LST
MCEBLD -- Building MCE .....
TKB @MCETKB
PIP MCE.TSK/PU/NM,MCE.MAP
PIP LB:[1,54]/NV/FO/CO=MCE.TSK
INS LB:[1,54]MCE
MCEBLD -- Procedure done
>
```