



2L FIELD DEVELOPER v 7.1

List of available CE commands in 2L

The commands are ordered by type. In the first column you will find the command and the required command-parameters. The second column contains a short description of the command. An asterisk (*) in front of the description means that the command is not implemented in the integrated test environment. You can not test this command during design.

A (f) at the end of the description indicates, the command can be used in form-mode only.

A (s) at the end of the description indicates, the command can be used in spreadsheet-mode only.

A (f+s) at the end of the description indicates, the command can be used both in form- and spreadsheet-mode.

Command type	Page
• File commands	2
• Field commands	3
• Search commands	4
• Test-condition commands	5
• Screenpage commands	6
• Form commands	7
• Spreadsheet commands	8
• (Error)Message commands	9
• Information commands	10
• Logfile commands	11
• Button commands	12
• Record commands	13
• List commands	14
• Commandfile commands	15
• Filterfile commands	16
• Timer commands	17
• Other commands	18



File commands



COPYFILE <oldfilename> <newfilename>

Copy file from <oldfilename> to <newfilename> (f+s)

DELETEFILE <filename>

Delete <filename> (without confirmation!!) (f+s)

DELETEFILE 0 <filename>

Delete <filename> if it exists and will not fail if <filename> does not exist

IFFILE <filename>

Test if <filename> exists (f+s)



Field commands

- EDIT ALL Make all data fields editable (f+s)
EDIT <fieldname> Make <fieldname> editable (f+s)
- SETFOCUS <fieldname>
 Make <fieldname> active input (f+s)
- NEXTFIELD Set active input (focus) to next input field (f+s)
- SAVE <filename> <value>
 Save <value> to <filename> (f+s)
- APPEND <filename> <value>
 Add <value> to <filename> (f+s)
- SET <value> <fieldname>
 Fill field <fieldname> with value <value> (Note, do NOT use any spaces in the value!!!) (f+s)
- SETVAL <fieldname> <valuestring>
 Fill <fieldname> with <valuestring> (f+s); <valuestring> can also be specified as a formula, e.g., %N2%+5 (only valid for numeric input fields).
- SETFIELD <fieldname>/<fieldindex> <valuestring>
 Fill <fieldname> (or indexed field, 0=first input-field etc.) with <valuestring> (f)
- SETACTFIELD <valuestring>
 Fill the active inputfield with Valstr (f)
- DRIVER <drivename> <filename> <fieldname>
 * Execute driver, get drivervalue from <filename> into <fieldname> (f+s)
- LOAD <filename> <fieldname>
 Load contents (first line) of <filename> into <fieldname> (f+s)
- LOADFROMREG <variable-name> <fieldname>
 Fill <fieldname> with the value from the registry-variable <variable-name> (f+s)
- SAVETOREG <Variable-name> <value>
 Save <value> in the handheld registry under variable-name (f+s)
- DISPWIDTH <fieldname> <width>
 Change the width of a specific <fieldname> to the <width> you specified (width in pixels) (f+s)
- AUTOCALC <mode>
 Sets AUTOCALC to <mode> (mode= ON or OFF). If autocalc is set to off the fields will not be recalculated continously but only at the end of the form, this saves you time. (f+s)
- RECALC Recalculate immediately (f+s)
- SUM <fieldname> <externalfieldname> <extdataset>
 Sums the value of the fields of <externalfieldname> from <external dataset> and fills <fieldname> with this sum
- SETMIDSTR <fldname> <startposition> <lgth> <valstr>



Fill <fieldname> with substring from <valstr> starting at <startposition> with <length> characters

SETSUBSTR <fieldname> <index> <valstr>

Fill <fieldname> with the <index> substring from <vlstr>, with space as separator

SETSUBVAL <fieldname> <index> <sep> <valstr>

Fill <fieldname> with the <index> substring from <vlstr>, with <sep> as separator

SETRANDOMVAL <fieldname> (<minval>) <maxval>

Fill <fieldname> with a random value between 0 (or <minval>) and <maxval>



Search commands

GETINDEX <searchstring>

Search record in present data set starting with <searchstring> (f+s)

GETNEXTINDEX <searchstring>

Find next record in present dataset starting with <searchstring> (f+s)

FINDVALUE <fieldname> <valuestring>

Search record in present dataset with value <valuestring> in field <fieldname>. If found go to this record (f+s)

FINDNEXTVALUE <fldname> <valstr>

Find next record in present dataset with value <valstr> in field <fldname>. If found go to this record. (f+s)

SEARCH

Start the search-procedure (as if the user selected the Search option from the Record-menu) (f+s)



Test-condition commands

- IF <condition>
Test if <condition> is correct. <Condition> is an equation like %N1%>12 or %N3%=>3.5 or %RECID%=%NEWID% (f+s)
- IFVAL<value> Test if value of active field is equal to <value> (f+s)
- EMPTY <value> Test if <value> (or %<fieldname>%) is empty (f+s)
- EMPTYFIELD <fieldname>
Test if contents of <fieldname> is empty (f+s)
- EMPTYLIST <fieldname>
Test if the list attached to <fieldname> is empty (f+s)
- NOTEMPTY <value>
Test if the <value> is NOT empty (f+s)
- NOTEMPTYFIELD <fieldname>
Test if contents of <fieldname> is NOT empty (f+s)
- NOTEMPTYLIST <fieldname>
Test if the list attached to <fieldname> is NOT empty (f+s)
- INLIST <listfield> <valstr>
Check if <valstr> is in list of <listfield> (f)
- IFFILE <filename>
Test if file <filename> exists (f+s)



Screenpage commands

-P	Go to previous screenpage (f)
+P	Go to next screenpage (f)
P <n>	Go to page <n> (f)
SHOW	Redraw actual page immediately (f+s)

Form commands

- EDITMODE <mode>
Set editmode for present dataset to TRUE (to enable editing) or FALSE (to disable editing for all fields) (f+s)
- FORM <formname> (<dataset>) (<searchstr>)
*Swap to other form/data set, start with record found via <searchstring> (f)
- FORM <formname> (<dataset>) (<searchstr>)
*Swap to other form/data set, without saving data (start with record found via <searchstring>) (f)
- +FORM <formname> (<dataset>) (<searchstr>)
*Swap to other form/data set with saving data (start with record found via <searchstring>) (f)
- RUNFORM <formname> (<dataset>) (<searchstring>)
*Run formmode as new task on top of the present form (swapping is faster, but requires more memory to run fast) (f)
- END Stop data-entry and go back to project-menu or start-form (f+s)
- CANCEL Stop data-entry without saving data, and go back to project-menu or start-form (f+s)



Spreadsheet commands

SHEET <formname> (<dataset>) (<searchstr>)

*Swap to other form/data set in spreadsheetmode (start with record found via <searchstring>) (f)

-SHEET <formname> (<dataset>) (<searchstr>)

*Swap to other form/data set in spreadsheetmode, without saving data (start with record found via <searchstring>) (f)

+SHEET <formname> (<dataset>) (<searchstr>)

*Swap to other form/data set in spreadsheetmode, with saving data (start with record found via <searchstring>) (f)

RETCOL

*Specify return column in spreadsheetmode (s)



(Error)Message commands

ERROR <errortext>

Show this error message (e.g. as cancelfunction) (f+s)

ERRORFILE <filename>

Show error message from <textfile> (f+s)

MESSAGE <messagetext>

Display <messagetext> in the message-line (at the top of the screen) (f+s)

ALERT <alerttext>

Pop-up an ALERT-box displaying the <alerttext> (f+s)

CONFIRM <confirmtext>

Ask the user to confirm <confirmtext>, continue if the user pressed OK, fail if the user pressed X (f+s)

BEEP <n>

Alerts the user. Is often used next to a error message.

Information commands

INFO Show standard info about actual form and record (like show info from the menu)
(f+s)

INFO NEW Clear the old info list (f+s)

INFO ADD <text>
Add <text> to the info list (f+s)

INFO ADDFILE <filename>
Add the contents of file <filename> to the info list (f+s)

INFO ADDFILE <filename> <starttext> (<endtext>)
Add parts of the contents of file <filename> to the info list. Start adding when
<starttext> is encountered, and stop adding when <endtext> is found (f+s)

INFO SHOW
Show the info list (f+s)

INFO HIDE
Hide the info list (f+s)

RECORDCOUNT <dataset> (<fldname>)
Return the number of records for the specified <dataset> in the variable _COUNT
(and the field <fieldname>)



Logfile commands

LOG <logfile> <value>

Add <value> (as new line) to <logfile> (f+s)

LOGFILE <textfile> <logfile>

Add the contents of <textfile> to <logfile> (f+s)



Button commands

KBCHAR	Your button acts as a keyboard key, using the first character of the description or name of the button (f+s)
KBSTR	The full description (or name) of the button will be used as value in the active field (f+s)

Record commands

- FIRST Go to first record in data set (f+s)
- PREV Go to previous record in data set (f+s)
- NEXT Go to next record in data set (f+s)
- LAST Go to last record in data set (f+s)
- GOTOREC (+/-)<recno>
Go to the specified record in dataset (if <recno> is specified with + or -, the jump to the new record is relative to the present record) (f)
- SAVERECNO (<filename>)
Save actual recordnumber to RECNO.INI or <filename> (if specified) (f+s)
- NEWREC Start data entry on a new record (f+s)
- SAVEREC Save present record to memory (f+s)
- ADDREC Copy present record as new record to data set (f+s)
- DELREC Delete present record (after confirmation) (f+s)
- GETREC <dataset> <searchcondition>
Get data from external <dataset>, where record matches <searchcondition>. A searchcondition is specified as fieldname=value (f)
- GETSREC <fnr> <fieldname>
Fill field <fieldname> with field field <fnr> (fnr starts from 0) from SEARCHREC
- GETSEARCHREC <fnr> <datafile> <searchstring>
Search <datafile> for first record that matches <searchstring> in fieldnr <fnr> (fnr starts from fnr 0) if found return this record in the SEARCHREC var.
- GETNEXTSEARCHREC <colidx> <dataset> <searchstring>
Search external <dataset> for next record that matches <searchstring> in column <colidx> (f+s)
able to use conditional search >12, >=88, etc.
able to use "reverse condition" 12<=
- SAVEDATASET
*Save data set to file (f+s)
- CHANGED <mode>
Set (<mode>: YES or TRUE) or release changes (<mode>: 0 or FALSE) to present record (f)
- GETINDEX <searchstring>
Search record in present data set starting with <searchstring> (f+s)
- GETNEXTINDEX <searchstring>
Find next record in present dataset starting with <searchstring> (f+s)
- SORT <sortcolumns> (<dataset>)
Sort the <dataset> on the <sortcolumns> (comma-separated list). If dataset is not specified, the active dataset will be sorted (and saved to disk) (f)
- SORT <collist> (<dataset>)



Sort the <dataset> on the <collist> (comma-separated list). If dataset is not specified, the active dataset will be sorted (and saved to disk) (f)

List commands

- LIST <textfile>
Display <textfile> as on-screen list (or hide this list) (f+s)
- LIST HIDE
Hide this list (f+s)
- LOADLIST <filename> <fieldname>
Load the contents <filename> into <fieldnamelist> (f+s)
- ADDTOLIST <fieldname> <value>
Add <value> to the <selection> list/stackfield <fieldname> (f+s)
- DELFROMLIST <fieldname> <mode>
Delete the selected line (or last line) from the <fieldname> list/stack, if <mode>= 0 :
do not ask for confirmation, if <mode>=1 : ask for confirmation, if <mode>= 99:
delete all items from the list (with confirmation) (f+s)
- SAVELIST <fieldname> <filename>
Save the items in the <fieldnamelist> to <file> (f+s)
- NOTEEMPTYLIST <fieldname>
Test if the list attached to <fieldname> is NOT empty (f+s)
- EMPTYLIST <fieldname>
Test if the list attached to <fieldname> is empty (f+s)
- GETLISTSTRING <index> <listfield>
Get <index> line from list attached to <listfield> into SEARCHREC. Fail if index is
not appropriate. (f)
- GETSELSTRING <listfield>
Get the selected item from the list attached to <listfield> and store in
SEARCHREC-record. Fail if no line was selected (f)
- INLIST <listfield> <valstr>
Check if <valstr> is in list of <listfield> (f)

Commandfile commands

CMD <filename>

Execute multiple commands from commandfile (f+s)

Note: the next commandfile commands are only valid in command files !!

DEBUG ON/OFF

Set debugmode ON or OFF (CMD) (f+s)

INCLUDE <filename>

Include commands from another <commandfile> into the active commandfile

REM <remark>

Put a remark line in your command-file (CMD) (f+s)

ONFAIL :Label

If one of the next commands in the command-file could not be completed, the command-file will be resumed from :Label (CMD) (f+s)

GOTO :Label

Go to the next :Label in commandfile (CMD) (f+s)

GOBACKTO :Label

Continue executing command file from a previous :Label

RESTART

Continue executing command file from the beginning

:Label

Label-line to be used with ONFAIL, GOTO and GOBACKTO commands (CMD) (f+s)

FAIL

Stop executing command immediately with a fail (CMD) (f+s)

Filterfile commands

FILTER <mode>

Set the filtermode ON or OFF (f+s)

SETFILTER <Fieldname> <Condition> <Value>

Filter a record on the condition(s) <Fieldname> <Condition> (which can be =, <=, => or <>) <Value>. Multiple conditions have to be seperated as condition1 | condition2 |...| conditionx (f+s)

FILTERDAT <fnr> <dataset> <filterdata> <condition>

*Filter <dataset> using <condition> on column <fnr> into <filterdata>

Note:the header of <dataset> will be copied to <filterdata>

FILTERTXT <fnr> <textfile> <filtertxt> <condition>

*Filter <textfile> using <condition> on column <fnr> into <filtertxt>

FILTERLST <fnr> <dataset> <filtertxt> <exportcolumns> <condition>

*Filter <dataset> using <condition> on column <fnr> into <filtertxt>. Only those columns as specified in <exportcolumns> (comma separated list) will be output, the header record will not be included.

Note: if condition is empty, all records will be used in the output.

Timer commands

Each can can have up to 10 user timers to periodic tasks when the form is active. Timeouts can be specified in seconds (default) minutes (e.g. 10m) hours (e.g. 1h). Any command or command-file can be used as <command>. Timers will automatically be stopped and deleted when the form or sheet is closed. (f+s)

SETTIMER <timer> <timeout> <command>

*Create a user timer (from 1 to 10), to run the <command> each <timeout>.

Note, SETTIMER doen NOT start the periodical execution of <command>. For this you need to call

STARTTIMER <timer> of STARTTIMERS.

STARTTIMER <timer>

*(Re)start timer <timer> now

STARTTIMERS

*(Re)start all user timer now

RESETTIMER <timer> (<newtimeout>)

*Reset timer <timer> to its original timeout (or to <newtimeout> if specified)

STOPTIMER <timer>

*Stop timer <timer> now, until STARTTIMER <timer> is called

STOPTIMERS

*Stop all timers now

KILLTIMER <timer>

*Stop and delete timer <timer> now. The timer cannot be started again, unless it has been recreated with SETTIMER.



Other commands

RUN <executable>

*Execute Batch-file or other program <executable> (f+s)

PW <password> (<inputtext>)

*Enter proper <password> to continue and you can add a <inputtext> above the field with the password(or exit start-form) (f)

RANDOM <maxvalue>

*Generate a random value between 0 and <maxvalue>. Store this value in variable
_RANDOM (f+s)

RANDOM <minvalue> <maxvalue>

*Generate a random value between <minvalue> and <maxvalue>. Store this value in variable
_RANDOM (f+s)

SETFIELDPOS <fldname> <newrow> <newcol>

*Move <field> (any datafield or button) on the fly to <newrow> <newcol> position.
Note this new position is lost when the form is re-opened. (f)

SETROWPOS <fldname> <newrow>

*Move <field> (any datafield or button) on the fly to <newrow> position.
Note this new position is lost when the form is re-opened. (f)

SETCOLPOS <fldname> <newcol>

*Move <field> (any datafield or button) on the fly to <newcol> position.
Note this new position is lost when the form is re-opened. (f)