

2L FIELD DEVELOPER v 11.X

USING VARIABLES AND REFERENCES

Most commands will take parameters where you can specify file names, field names or values.

1. SETVAL

SETVAL Fldname Valstr

SETVAL is the name of the command (in capitals !!).

Fldname must be the name of one of your data fields referenced in the form e.g. N2, WEIGHT or NAME.

Valstr can be any value or text e.g. 12, 25.75 or John.

Note: If you reference a Numeric field, and specify a non-numeric value, the value will not be accepted, and the command will fail.

Instead of a literal value for Valstr, you can use the contents of another data field or internal variable, by using %% signs around the field name. So SETVAL could be specified as SETVAL N3 %WEIGHT %

This last command will try to fill data field N3 with the actual value of data field WEIGHT.

2. SAVE and LOAD

SAVE Filename Valstr

LOAD Filename Fldname

SAVE/LOAD is the command name (in capitals).

Filename is the name of a file in your project folder.

Valstr can be any numeric value or text, or a reference to a data field.

Fldname must be the name of one of your data fields referenced in the form e.g. WEIGHT, A4 or COMPANY.

The LOAD command will load the first line of text from the contents of the Filename into the data field Fldname. It will fail when Filename does not exist.

The SAVE command will save the the actual value of Valstr in the file Filename. An existing file will be overwritten. This command will fail if it could not create the file Filename.

These commands can be used for exchanging information over records or forms.

SAVE TEMP.TMP %WEIGHT%

This will save the contents of data field WEIGTH to a file (in your project) called TEMP.TMP.

Now you can access this value from another form or record with a command like LOAD TEMP.TMP W2.

This will load the contents of the file TEMP.TMP into the data field W2.

3. Registry variables

Instead of using files to transfer data from one form, record or dataset to another form, record or dataset you can also use Registry variables, with names you define yourself. Using registry variables you can even transfer data from one project to another project.

There are 2 specific commands for registry variables SAVETOREG and LOADFROMREG, but registry variables can also be referenced in most commands (like SETVAL or LOAD) instead of data fields.

These registry variables are referenced using %% signs and a _REG_ prefix in front of the name of your registry variable. For example:

```
SAVETOREG C12 12
```

This command will create (or overwrite) a registry variable called C12 with value 12

```
SETVAL N3 %_REG_C12%
```

This command will try to fill data field N3 with the value from the registry variable C12.

You can also use the registry prefix in a SETVAL command:

```
SETVAL _REG_C13 %_REG_C12%
```

This last command is equivalent to SAVETOREG C13 %_REG_C12%

4. Other uses of variables and references

Variables and references not only can be used in commands, but also in field descriptions and even in Text and Picture names or as button descriptions.

Like:

Number of %NAME%

Weight of %NTOT% fruits

%PICTURE%.bmp

Or use them in formulas together with a SETVAL command:

```
SETVAL AVGWEIGHT (%GROSWEIGHT%-%CRATEWEIGHT%)/%NUMBER%
```

5. List of available program variables

2L uses a number of so-called program variables. With a few exceptions they all have an underscore (_) prefix.

===== PROJECT RELATED variables =====

_PROJECT

the name of the active project

_MODULE

the name of the active module (can be empty if no module in use)

_FORMNAME

the name of the active form

_FORM

the name of the active form

_DATASET or _DATASETNAME

the name of the active dataset

_RECNO

the actual record number in the active dataset

_RECCOUNT

the number of (filtered) record of the active dataset

_FIELDVAL or _FIELDVALUE

the actual value of the present input field

_FIELDNAME

the name of the present input field

===== DATE variables =====

_DD or _DAY
the day formatted as DD, ranging from 01 to 31

_MM or _MONTH
the month formatted as MM, ranging from 01 to 12

_YY
the last 2 digits of the year, ranging from 00 to 99

_YYYY or _YEAR
the year in 4 digits

_DDMM
day + month formatted as DDMM

_WEEK
the present week number, ranging from 01 to 53

_DATE
the date formatted using the default language related date format (MM/DD/YYYY or DD-MM-YYYY) or the user specified DATEFORMAT

_DDMMYY
the date formatted as DDMMYY

_MMDDYY
the date formatted as MMDDYY

_YYMMDD
the date formatted as YYMMDD

_YYYYMMDD
the date formatted as YYYYMMDD

_DDMMYYYY
the date formatted as DDMMYYYY

_MMDDYYYY
the date formatted as MMDDYYYY

_DATEFORMAT
the actual DATEFORMAT specification

===== TIME variables =====

_HH
the hour formatted as HH, ranging from 00 to 23

_HHMM
the time in hour + minutes formatted as HHMM

_HHMMSS
the time formatted as HHMMSS

_TIME
the time formatted as HH:MM:SS

_DATETIME
the combined date + time formatted as DATEFORMAT TIMEFORMAT

_TIMEFORMAT
the actual TIMEFORMAT specification

Note:

To fill date/time fields with date or time stamps or to add date or time to your own log-files you can use date and time variables between % % signs. For example:

```
SETVAL MYDATE %_DATE%  
LOG MYLOGFILE %_YYYYMMDD% %_HHMMSS%
```

===== other variables =====

_2LVERSION
the version of the 2L program

_2LDIR
the full path to the 2L Field program folder



_PROJECTSDIR
the full path of the 2L Projects folder

_PROJECTPATH
the full path of the active project (includes projectdir, module and project folder)

_DRIVERDIR
the full path to the Drivers folder

_BACKUPDIR
the full path to the Backup folder

_SVGDIR
the full path to the SVG folder

_HHID
The unique handheld ID (Serial number)

_COUNT or **_CALCCOUNT**
The number of records from the last RECORDCOUNT command

_SUM
The value as result of the last SUM command

_RANDOM or **_RANDOMVAL**
the last random value generated with the RANDOM command

REG<yourownregistryvariable>
If you've set your own registry variable MYREG using the command
SAVETOREG MYREG test
the variable %_REG_MYREG% will now hold the text test.

===== other variables WITHOUT underscore prefix !!!! =====

SEARCHREC
the record content from the last GETSEARCHREC/GETNEXTSEARCHREC command

SEARCREC.n
the n-th field of the SEARCHREC variable