

Documentation

Leibinger Interface protocol

For printer series

JET₂neo

JET3

JET3^{UP}

Version 1.9.3

Summary

General.....	4
Validity.....	5
Script Processing.....	6
A script starts with.....	6
and ends with.....	6
Action instructions	7
Overview	7
Details to commands.....	7
!RC	7
!LF.....	8
!US	8
!EQ	8
Inquiries.....	10
Parameter Transfer	11
=AC All Counter (Machine → Host)	11
=CC Current Counter (Machine ↔ Host).....	11
=CM Config Mail (Machine ← Host).....	12
=CO Change Output (Machine ← Host).....	12
Attention.....	12
=DS Available Flash memory (Printer → Host)	13
=EL ErrorList (Machine → Host)	13
=ET ExternText (Machine ← Host)	13
=EV EventList (Machine → Host).....	15
=EX Extern States (Machine → Host).....	15
=FC CRC32 failed (Machine → Host)	17
=FL Fix Blocklength (Machine ← Host).....	17
=GW Wysiwyg Data (Machine → Host)	17
=HS Hydraulic State (Machine → Host).....	17
=IC Input Configuration (Machine → Host)	18
=JL JobLoad (Machine ↔ Host)	19
=MC Main Configuration (Machine → Host).....	19
=MR Mail Record (Machine ← Host)	20
=MS Maschine status (Printer → Host)	21
=NR Next CRC32 (Machine ← Host).....	22
=OC Output Configuration (Machine → Host).....	22
=PR Print Parameter (Machine ← Host)	23
=RC Reset Counter (Machine ← Host).....	23
=RS Machine State Information (Machine → Host)	23
=SC Service Configuration (Machine → Host).....	25
=SH Service History (Machine → Host)	26
=SM Mail Status (Machine → Host).....	26
=SW Start WYSIWYG (Drucker← Host)	27
=VS Version (Machine → Host)	27

File Transfer	28
\$RD Request-Directory (Machine ← Host)	28
\$DI Directory-Info (Machine → Host)	29
\$RF Request File (Host → Machine)	29
\$FH File Header (Host ↔ Machine)	30
\$FT File Transfer (Host ↔ Machine)	30
\$FA File transfer-acknowledge (Printer ↔ Host)	30
Annex A Transmissions with CRC32.....	31
Annex B Mailing (Database) operation.....	33
History	41

General

The interface allows remote control and remote diagnosis access via Ethernet, RS232 and modem. The interface is based on a dual-master protocol which is able to send from both sides independently.

The protocol generates a frame around the LJscript-language so that the interface can directly transfer script-sequences.

To offer a terminal-function the protocol works without a direct package reply. For the usage of a serial interface it is required to activate the handshake-controlled hardware-flow control in the transmission range to avoid a receive-sided overflow.

All packages are started with a ^-character and are finished with the character <CR> (Hex 0D). Optionally you can add a <LF> (Hex 0A).

After the start character the destination address follows, which consists of a character in the range of 0-9 and A-F. To address a LJ3-Printer you have to use the address "0".

Further addresses are not used at the moment, they are just used as reserve for later options. Afterwards a character continues which selects a command group. These are:

Character	Command-group
*	Script-instruction (a line continues as described in LJ-script)
!	Action-instruction (Printer control, e.g. nozzle open/closed, Print start, test print etc.)
?	Inquiry (The designation of the instruction which has been inquired continues e.g. ?HS inquires the current hydraulic-settings)
=	Parameter Transfer (e.g. =HS7100 2400 is the response of the machine to ?HS inquiry)
\$	File Transfer, Directory Info

The transmitted data within the frame are ASCII-data (8 Bit, 0x00...0xff). Basically all values are allowed but some values have special functions:

Character	
0x00	Will be replaced by blank characters (0x20).
0x0d	End of command. Is not allowed within data. Empty commands (more 0x0d in succession) are allowed

The handling of further special characters is described in the following chapter.

Special characters

The data transferred within a frame are ASCII data which must not include the following codes: ^ (0x5e) and <CR> (0x0d). In case the ^ character shall be transferred as data it has to be marked with a preceding backslash „\^“. A backslash character to be transferred has to be doubled (\\). Variables if not defined otherwise are passed over as tab separated decimal values.



Exceptions:

JET3, JET2neo up to version V71.0.27.00 and the WinJet software up to version V3.0.3.1 do not recognize special characters at all. In these cases special characters must not be used in mailing records, scripts or file names!

Validity

The following definitions of the commands are only valid for the turned on device. If the device is in standby mode, only the instruction !PO (Power ON), !RM (Reset) or FileTransfer can be handled.

The command list of the parameters may be extended in later versions. That means each receiver has to be prepared to receive more data and it must be able to ignore excess or unknown data.

Script Processing

The instructions which are described in the LJ-script can be transferred directly with Frame-Header.

A script starts with

`^0*BEGINLJSCRIPT<CR>`

and ends with

`^0*ENDLJSCRIPT<CR>`

As soon as the instruction BEGINLJSCRIPT has been transmitted, the current job data file (that means the work file which is loaded during the reboot of the system) is deleted and will be overwritten with the following script instructions.

As soon as the ENDLJSCRIPT instruction has been transmitted the data file is passed to the RIP for printing. In this situation ****EXTERN**** is displayed as job name.

<CR> is the character Carriage Return and has the value Hex 0D (0x0D).

Action instructions

Action instructions are commands which are transferred without parameters and which carry out a direct reaction in the printer. The following actions are implemented.

Overview

Command	Description
!GO	Start Print-Ready
!ST	Stop Print-Ready
!NC	Nozzle Close
!NO	Nozzle Open
!RC	Reset Counter (Object Counter)
!FA	Factory Reset
!MC	Motor Step Close
!MO	Motor Step Open
!LF	Re-Load Fonts
!OK	CRC OK (Host←Printer)
!US	User Suspend (Lock of printer operation)
!UR	User Resume (Release of printer operation)
!EQ	Error Quit
!PO	Power ON (Out of Standby)
!PF	Power OFF (in Standby mode)
!EN	End of Data (Host←Printer)
!RM	Restart Machine (= Software Reset, only while Standby)
!W0	Wysiwyg display off
!W1	Wysiwyg display on
!FF	Flush Mail Fifo (Clearing the Mailing FIFO)
!LN	The LJ3 sends in the protocol the length
!EM	Switch on the echo mode

Details to commands

!RC

Reset all counters to its initial value which are defined in a job as counter reset source. This function has no effect on the product counter (for settings of product counter please see =CC Current Counter (Machine ↔ Host))

Example: ^0!RC 0x0D

See also command =RC Reset Counter (Machine ← Host)

!LF

This instruction is necessary if fonts have been changed or added and which should be used by the current job.

Attention! This instruction causes a short interrupt of the print out.

!US

This instruction prevents user - inputs for 10 seconds (information window on the operator interface opens).

If the interface should be blocked for the operator for the duration of a remote connection, the remote application has to repeat this instruction within 10 seconds continuously or a communication should be proceeded within 10 seconds (inquire status, read files etc.) that means the user can operate the device again after activation of the suspend mode and a communication break of >10 seconds. This function should provide an operation even in case of a connection abort. By sending the !UR command you can leave this mode and the operation will be enabled immediately.

!EQ

Provides the acknowledgement of arising errors at the machine.

Errors have to be acknowledged (especially in the User Suspend mode) because the machine will turn off after a certain period of time due to safety reasons. If and which errors are occurred can be determined with ?RS (see there).

!LN

The LJ3 sends from this moment in his protocol the length of data. This will be done until the LJ3 will be switched off or he gets a factory reset command.

The protocol is not changed for the receiving of the Leibinger-Jet3. Here the length will not be sent.

The length is always a decimal number with 5 digits and comes directly after the address byte and before the command group character.

This length defines the number of bytes which are coming from now on including the terminating carriage return.

Example:

Host sends:	LJ3 answers:
^0?RS <CR>	^0=RS2 6 0 0 0 <CR>
^0!LN <CR>	
^0?RS <CR>	^000013=RS2 6 0 0 0 <CR>

!EM

Switch on the echo mode. This remains switched on until the LJ3 will be switched off or he gets a factory reset command.

Function: The printer sends all action- and parameter commands directly after the receiving back to the host. The command will only be processed by printers which are

able to recognize the command. These are the commands with the ! and = in the command group.

Inquiries

With an inquiry you can demand parameters from the opposite side.

The opposite side responds on an inquiry with the according =XX parameter command.

Overview of inquiries (Host → Machine)

Command	Description
?AC	Query of all counters in the current job. (see also =AC All Counter (Machine → Host))
?CC	Query product counter reading
?DL	Query date/time
?DS	Query free internal flash memory space
?EL	Query error list
?ET	Query current Extern Text
?EV	Query current event list
?EX	Query extern states
?GW	Query current WYSIWYG
?HS	Query hydraulic status
?IC	Query configuration of the input signals
?JB	Query current print job (not the name but the script!) The printer will send the current print job in ^0* BEGINSSCRIPT... blocks. See also Script Processing)
?JL	Query name of the currently loaded print job
?MC	Query basic configuration
?MS	Query machine state (see also =MS Maschine status (Printer → Host))
?OC	Query configuration of the output signals.
?PR	Query of individual job parameters (see also =PR Print Parameter (Machine ← Host))
?RS	Query status
?SC	Query service configuration
?SH	Query service history
?SM	Query mailing status
?VS	Query software version

Normally queries do not have any parameters. There are the following exceptions:

Command	Additional Parameter
?PR	Parameter to specify the job parameter in question. (see also =PR Print Parameter (Machine ← Host))

Parameter Transfer

With the help of parameter-transfer commands you can send current settings and status-information from the machine to a host or new settings can be transferred from host to the machine. The single parameters are separated with <TAB>-characters. In case a parameter is empty, the current value will not be changed. The following parameter-instructions are implemented:

=AC All Counter (Machine → Host)

Par1: Number of counter used in the current job

The maximum number of counters depends on the type of printer. Please see the documentation of the script language for details.

For each counter there are 3 parameters:

Counter n:

Par $3*(n-1)+2$: Done repetitions

Par $3*(n-1)+3$: Counter value High

Par $3*(n-1)+4$: Counter value Low

Parameter Details:

Done repetitions

This is the value how often this counter value was printed. When the counter parameter "repetitions" is 0, this value is always 1.

Counter value High and Low

Because a counter can have up to 10 figures, there are 2 32 bit values necessary.

=CC Current Counter (Machine ↔ Host)

Par1: Current reading product Counter

Par2: Stop after X Products

Par3: Total print counter

Stop after X Products

With reaching this product counter status the print is stopped automatically by the LJ3. (Exception 0 = function deactivated)

Total print counter

The total print counter can only be read (read only) and not be changed.
 The counter reading is the number of printouts of the printer.

=CM Config Mail (Machine ← Host)

Par1: StopAtNo

After sending this parameter the printer stops the print automatically if this mail record number has been printed.

Exception is 0 (default). No print stop is carried out for 0.

For the query of this parameter see ?SM

=CO Change Output (Machine ← Host)

Par1: No. of the output which should be changed. Range of value: 0...8

Par2: Wished state of the output. Range of value: 0 or 1

The printer has the outputs 1...8.

Special case: Output No. 0:

^0=CO0 1: Command Initializing

^0=CO0 0: Command End. The Jet3 now controls the outputs itself again. All external changes will be undone

Before with this command could be worked, at first one time the command "Initializing" should be sent. Only in this case the settings made in I/O Settings for these outputs believe conserved for which no CO command will be sent.

In order to avoid conflicts, the outputs which will be changed with =CO, should not be used in the I/O Settings.

Attention

With the external control of the outputs either by initialization or by setting an output, the outputs will be changed by the printer anymore. This means there will be no more warning, error or trigger signals generated by the printer during with the =CO command.

Example:

^0=CO0 1 Initializing.

^0=CO3 1 Set output 3 to 1. Output „Warning“ is not available.

...

^0=CO0 0 End. Output 3 and „Warning“ are in the original state again.



This command is not available for JET2neo printers

=DL Set Date/Time (Machine ↔ Host)

Parameter	Value	Format
1	Year	1601.. 30827
2	Month	1..12; 1 = January
3	Weekday	0..6; 0 = Sunday
4	Day	1..31
5	Hour	0..23
6	Minute	0..59
7	Second	0..59

All parameters are required!

Example:

^0?DL

^0=DL2015 10 1 26 7 45 23

=DS Available Flash memory (Printer → Host)

Results in the available flash memory in kByte

=EL ErrorList (Machine → Host)

The error list is transmitted in blocks. Each block contains the following parameters separated by tabs:

Par1: Number of first entry (1...)

Par2: Total number of entries

Par3: Number of entries in this block

Par4: Entry1

Par5: Entry2

Each entry is transmitted using the following pattern:

yy-mm-dd hh:mm:ss/errornumber

The maximum number of entries for each block is 32.

=ET ExternText (Machine ← Host)

Par1: String

String = External text with up to 2048 characters and <CR> for final identification
Reaction time of the printer: 20...40msec

When the job of the printer contains a graphic which uses ExternText, the string should contain a graphic at the corresponding position. The format of the graphic is described in the documentation of the script language.
See chapter OBJ parameter list

Text-example:

`^0=ETHallo0x0D`

Grafic-example:

`^0=ET(8 3 FF81FF)0x0D`

`^0=ETHallo(8 3 FF81FF)0x0D`

=EV EventList (Machine → Host)

The event list is transmitted in blocks.. The EventList contains:

Par1: Number of the first entry (1...)

Par2: Total number of entries

Par3: Number entries in this block

Par4: Entry1

Par5:

Each error entry is transmitted in the form

yy-mm-dd hh:mm:ss/cp/ss/st/mr/pr/dp/ft/ms

with

yy-mm-yy	Date
hh:mm:ss	Time
cp	Compressor-Pulse during the last minute
ss	Sips of solvent during the last minute
st	Sips of ink during the last minute
mr	Mixing ratio. Number of sips of ink before solvent was added
pr	Pressure of the pressure tank in mbar
dp	Diaphragma-position in the pressure tank. Non-dimensional, typically between 700 and 1400.
ft	Fall time of the visco-meter in milliseconds
ms	Machine status. Please see =RS Machine State Information (Machine → Host) for details.

Each block has a maximum of 16 entries

=EX Extern States (Machine → Host)

Par1: Actual Speed

Par2: Inputs (Integer number from 0 – 2¹⁷)

Par3: MaxSpeed

Par4: Direction

Par6: PrintGo

Par7: Reserved

Par8: Reserved

Par9: Reserved

Parameter details:

Actual Speed:

Information of the current printing speed in decimeter/min



Attention:

JET3 versions up to V60.0.1.17: in meter/min

JET2neo versions up to V71.0.00.11: in meter/min

Inputs:

Bit 0 = X4.9
Bit 1 = X4.22
Bit 2 = X4.10
Bit 3 = X4.23
Bit 4 = X4.11
Bit 5 = X4.24
Bit 6 = X4.2 (Counter Increment)
Bit 7 = X4.1 (Counter Reset)
Bit 8 = X4.3 (Job select 0)
Bit 9 = X4.16 (Job select 1)
Bit 10 = X4.4 (Job select 2)
Bit 11 = X4.17 (Job select 3)
Bit 12 = X4.5 (Job select 4)
Bit 13 = X4.18 (Job select 5)
Bit 14 = X4.6 (Job select 6)
Bit 15 = X4.19 (Job select 7)
Bit 16 = X4.7 (Job select 8)
Bit 17 = X4.20 (Job select 9)

Max Speed:

Maximum possible printing speed for current setting in decimeter/min.

**Attention:**

JET3 versions up to V60.0.1.17: in meter/min

JET2neo versions up to V71.0.00.11: in meter/min

Direction:

Encoder direction:

0 = Backward

1 = Forward

PrintGO signal level

0 = PrintGo Sensor Signal is Low

1 = PrintGo Sensor Signal is High

Reserved:

Always 0

**Attention:**

The JET2neo interprets Jobselect signals only up to Jobselect 7.
Here X4.7 und X4.20 are reserved.

=FC CRC32 failed (Machine → Host)

Par1: CRC32 Checksum which has been calculated by the machine and which did not correspond with CRC32 transmitted with =NR.

=FL Fix Blocklength (Machine ← Host)

Par1: Block length in bytes

With this command the printer is adjusted to send with a fix block length.

All data, which the LJ3 sends, will be filled up to this block length with blanks.

The block length can be adjusted in the range between 1 and 2048. It will be active until the next power on of the printer.

=GW Wysiwyg Data (Machine → Host)

The response happens block by block with max. 60 characters per line.

For final identification the sender (machine) transfers a ^O!EN.

Data layout:

Byte 1-16	Header information:
Byte 1-4	Overall width of bitmap
Byte 5-8	Height of bitmap in pixel
Byte 9-12	Job ID of this bitmap (for ext. Jobselect the applied ext. number)
Byte 13-16	Number of actual delivered strokes. The current picture may be smaller if the printer for performance reasons shall delivers only a detail of the picture. In that case the size depends on the type of printer The offset of the detail is defined with the command =SW Start WYSIWYG (Drucker ← Host)
Byte 17 till end	Stroke data

60 characters per line, nibble coded that means 30 bytes per line

Coding High Nibble (Bit 7-4) 0-15 = A-P

Coding LowNibble (Bit 3-0) 0-15 = A-P

=HS Hydraulic State (Machine → Host)

Par1: ActFalltime

Par2: ActPress

Par3: ActSuctpump
Par4: Diaphragm Position
Par5: Mix-Value
Par6: State Current status

Parameter details:

Act Falltime:

Current fall time of the viscometer in ms.

ActPress:

Current pressure in mBar

ActSuctpump

Current suction pump value in %

Diaphragm position

Current diaphragm position of the pressure tank.

Dimensionless, normal operating positions between 700 and 1400)

Mix-Value

Current solvent mix-value in ink to solvent.

0 = no solvent admixture

State:

Not documented

=IC Input Configuration (Machine → Host)

Par1: Input1 (assigned X4.9 function)
Par2: Input2 (assigned X4.22 function)
Par3: Input3 (assigned X4.10 function)
Par4: Input4 (assigned X4.23 function)
Par5: Input5 (assigned X4.11 function)
Par6: Input6 (assigned X4.24 function)

Parameter details:

Decoding of numbers to the assigned function:

0 = Disabled
1 = Nozzle Open
2 = Nozzle Close
3 = PrintStart
4 = PrintStop
5 = Printer Shutdown
6 = Horizontal Mirror
7 = Cutting Synchronization
8 = Counter reset 1
9 = Counter reset 2

10 = Load job
 11 = Ink on/off
 12 = Batch job reset



Note:
 Not all functions are available with all printers.
 The JET2neo provides only 3 inputs!

=JL JobLoad (Machine ↔ Host)

Machine->Host: The Ljet sends the current Jobname with path.
 Host->Machine: The Ljet loads the job with the name in Par1.
 Reaction time of the LJ3: 50...150msec

Par1: Jobname (with path information)

Example: ^0=JLFFSDISK\JOBS\TESTPRINT.JOB 0x0D

=MC Main Configuration (Machine → Host)

Par1: DirectPrint
 Par2: Encoder Setting
 Par3: Reserved (0)
 Par4: Reserved (0)
 Par5: Reserved (0)
 Par6: Set Fall Time
 Par7: Set Pressure
 Par8: Reserved (0)
 Par9: Set Visco Correction
 Par10: System Frequency
 Par11: Inknumber
 Par12: Interval Time on
 Par13: Interval Time off

Parameter Details:

DirectPrint:

0 = Direct Print function deactivated / 1 = Direct Print function activated

Encoder Setting:

0 = RS422
 1 = TTL (5V level)
 2 = HTL (24V level)

Set Fall Time

Current set target fall time of the viscometer in ms

SetPressure

Current set system pressure in mbar

SetViscoCorrection

Current set viscometer correction factor

SystemFrequency

Dot frequency in kHz

InkNumber

Ink-Number which is filled in the printer.

Interval Time ON / Off

Interval times in the following format:

Hour*100+minutes

Hours 0-23 and minutes 0-59

=MR Mail Record (Machine ← Host)

Par1: String

The String can have a length up to 2048 Bytes. When he is longer he will be truncated from the printer.

The string is assembled as follows (e.g. 3 Field Record):

No + tab + Field1 + tab + Field2 + tab + Field3+ ... + tab + {[h][ox][rn]} 0x0D

Example:

^0=MR17 Miss Jane Miller {ho2r3}0x0D

Definitions:

No. is a displayed ASCII consecutive 32 Bit No. without any prepend zeros and has to exist coercively!

Field should not contain ^ or has to be replaced by \^ !

A record can contain up to 255 fields.

Enhancements:

JET3 from V60.0.1.22

JET2neo from V71.0.22.00

When in the job of the printer contains a graphic which uses mailing, the corresponding text field should contain a graphic. The format of the graphic is described in the documentation of the script language. See chapter OBJ parameter list.

Example:
 $\wedge 0 = MR0 \quad 111 \quad (8 \ 3 \ FF81FF) \ 0x0D$

The mail record may contain **Property-Parameter** in curly brackets: {}. These are after a tab directly before the 0x0D.

The square brackets [] are not used. They have the meaning that parameters within these brackets are optional. Each of these 3 parameters may be used only once.

h = horizontal mirroring of this record
 o0 = Orientation 0°
 o1 = Orientation 90°
 o2 = Orientation 180°
 o3 = Orientation 270°
 rn = n repetitions of this record

The parameter repetition may only be used when the first parameter, the serial number, is always 0.

Examples:

$\wedge 0 = MR0 \quad \text{text1} \quad \{ho2r3\}0x0D$
 $\wedge 0 = MR0 \quad \text{text2} \quad \text{text3} \quad \{r21\}0x0D$

=MS Maschine status (Printer → Host)

Parameter 1	Actor state compressor
Parameter 2	Actor state main pump
Parameter 3	Actor state suction pump
Parameter 4	Actor state visco pump
Parameter 5	Actor state air outlet valve
Parameter 6	Actor state bleed valve
Parameter 7	Actor state ink flow valve
Parameter 8	Actor state mixing valve
Parameter 9	Head motor state (nozzle seal) 1 = opening 2 = opened 3 = closing 4 = closed 5 = inbetween
Parameter 10	Printing is active
Parameter 11	Reserved/Test

=NR Next CRC32 (Machine ← Host)

Par1: CRC32 Checksum of the proximate communication string.

=OC Output Configuration (Machine → Host)

Par1: Output1 (signal assigned to X3.1)
Par2: Output2 (signal assigned to X3.2)
Par3: Output3 (signal assigned to X3.3)
Par4: Output4 (signal assigned to X3.4)
Par5: Output5 (signal assigned to X3.9)
Par6: Output6 (signal assigned to X3.10)
Par7: Output7 (signal assigned to X3.11)
Par8: Output8 (signal assigned to X3.12)

Parameter Details:***Decoding of No. to the assigned signal***

0 = Disabled
1 = Refill Warning
2 = Error/Print Stop
3 = Print Readiness
4 = Print active
5 = Product counter reached end value
6 = PrintGo
7 = Encoder Signal (left/right)
8 = Power fail
9 = Camera trigger
10 = Counter overflow
11 = Camera error
12 = Camera warning
13 = Maximum speed reached

Hints:

- The encoder signal may have the value „1“ or „0“. Which one is left or right depends on the way the encoder is installed!
- The error signal for power fail is inverted. The parameter has the value “1” as long as the power supply of the printer is working.



**Not all printer models support all functions.
The JET2neo has only 4 output signals.**

=PR Print Parameter (Machine ← Host)

since V60.0.01.04D

Par1: Print parameter type, which could be changed

Par2: New value for the parameter

With this command a single parameter in the Jet3 can be changed.

Par 1:	Meaning of Par1:	Unit of Par 2:	Time of the result:
1	Print Go delay	Mikrometer. 1mm=1000µm	With the 2. PrintGo
2	Print Go distance	Mikrometer. 1mm=1000µm	With the 2. PrintGo
3	Strokedistance	Mikrometer. 1mm=1000µm	With the next PG
4	Printing hight	30 ... 100	With the next PG
5	Speed of the internal encoder pulses	In dm/min *)	With the next PG

Example:

^0=PR2 31000 Sets the PrintGo distance to 31mm (31000µm)



*) Attention:

JET3 versions up to V60.0.1.17: meter/min

JET2neo versions up to V71.0.00.11: meter/min

=RC Reset Counter (Machine ← Host)

Par1: ID-Number of the counter

Par2: Start value

Reset counter means that the counter will be set to the start value.

Par1 contains the ID-Nr. of the counter, which have to be reset. If Par1=0 all counters are reset.

With Par2 you can send the start value. If this value is -1 the start value will be the start value of the job. Otherwise the counter will be reset to this parameter. If no Par2 is given, Par2 = -1(Job setting)

=RS Machine State Information (Machine → Host)

Par1: Nozzle state

Par2: State of Machine

Par3: Error number

Par4: Head cover

Par5: Current Speed

Par6: Flag for job change

Definitions:

Nozzle state:

0 = invalid (Standby or initialization)

1 = opens

2 = is open

3 = closes

4 = is closed

5 = inter, in between

State of Machine

1 = Standby (The other values e.g. nozzle state are invalid !)

2 = Initialization phase (incl. Bleeding)

3 = Interval or Service Panel

4 = Ready for Action (ready for operation, instructions as e.g. nozzle open can be carried out)

5 = Ready for Print start

6 = Printing

Error:

0 = no error

xxx = Error number

Besides the error code further information regarding the error source and malfunction are defined in the error number:

Bit No.	Meaning	Values
26/25	Error source	00 = FEP CPU. (User interface) 01 = RIP CPU (Print processing + hydraulic) 10 = SDC CPU (Dots + Print head control)
27	Malfunction Shutdown	0 = Device turns off after 30min. if no confirmation happens. 1 = Device turns not off
29/28	Malfunction Signal tone transmitter	00 = Permanent signal tone 01 = one-time signal tone 10 = no signal tone
31/30	Malfunction Display	00 = Display error window 01 = Display warning window 10 = Display message window

After masking bits 25-31 one error code remains which specifies the occurred error. A list of error texts is available on request.

Headcover

0 = Headcover closed
1 = Headcover open

Current Speed

In m/min (Vor V60.0.1.17: meter/min)

Flag for job change

The printer software sets a **Flag for job change** in case there were made changes within the print job or the print job itself was changed since the last status query.
0 = no changes, 1 = changes occurred

=SC Service Configuration (Machine → Host)

Par1: Service Hour Interval
Par2: Service Year Interval

Parameter Details:

ServiceHourInterval:

Operating hours between two service cycles

ServiceYearInterval:

Days between two service cycles

=SH Service History (Machine → Host)

Par1: Operating Hours

Par2: Last Service Hours

Par3: Last Service Date

Parameter details:

Layout Last Service Date (32 Bit):

Bit 31-26: Year-2000

Bit 25-22: Month (1-12)

Bit 21-17: Day of month(1-31)

Bit 16-12: Hour (0-23)

Bit 11-6: Minute (0-59)

Bit 5-0: Second (0-59)

=SM Mail Status (Machine → Host)

Par1: Fifo Depth

Par2: Fifo Entries

Par3: Last printed record number

Par4: Record number after which the printout is automatically stopped

Par5: Last printout was finished

Description of parameters:

Fifo Entries	Number of mail records which are available in the mailing Fifo. The number of FIFO entries is always smaller by 1 than the number of records actually sent. The reason for this is that the currently printed record was already substructed.
Record number after which the printout is automatically stopped	Number of the mail record which is currently printed. If no record is available the value is 0.
Stop record number	Record number at which a print stop is generated. This value was set with ^0 = CM.
Last printout was finished	Flag. If = 0, the last print was started, but not finished, if >0, the last print was finished, but the next one for sure not started



To be sure that these are the most current parameters it is mandatory the host sends the command ^0?SM to the printer twice . The second answer of the printer shows definitely the most current parameters.

=SW Start WYSIWYG (Drucker ← Host)

Defines the starting point in x-direction at which the WYSIWYG data inquired with the ?GW command will be sent. With this it is possible to scroll the WYSIWIG screen.

Parameter 1: X-Start

=VS Version (Machine → Host)

Par1: FPGA-Version

Par2: RIP-Software Version

Par3: SDC-Software Version

Par4: FEP-Software Version

Par5: KernelVersion

Par6: SystemType

Par7: LanguagePack

Par8: FontPack

Par9: LJVersion of firmware

Par10: Type

Par11: Serialnumber

Parameter details:

All parameters are tab separated strings

File Transfer

In the FileTransfer-area data and directories are exchanged. The following instructions are supported:

Standard directories of the printers

Data carrier	Memory type	Path
Internal Flash Disk		
JET2neo JET3 JET3up	Jobs	FFSDISK\Jobs
	Fonts	FFSDISK\Fonts
	Graphics	FFSDISK\Graphics
SD Card		
Only JET3	Jobs	Storage Card\LJ3\Jobs
	Fonts	Storage Card\LJ3\Fonts
	Graphics	Storage Card\LJ3\Graphics
USB Stick		
JET3 JET3up	Jobs	USB-Stick\LJ3\Jobs
	Fonts	USB-Stick\LJ3\Fonts
	Graphics	USB-Stick\LJ3\Graphics
USB Stick		
JET2neo	Jobs	USB-Stick\LJet2neo\Jobs
	Fonts	USB-Stick\LJet2neo\Fonts
	Graphics	USB-Stick\LJet2neo\Graphics

\$RD Request-Directory (Machine ← Host)

Sends a directory inquiry to the machine.
The machine responds with a DI-info.

Par1: Directory string

Within the directory string also the common substitution characters are valid (Joker *)

Some examples:

Check existence of the job directory:

Inquiry: ^0\$RDIFFSDISK\Jobs

Response: ^0\$DI1 01 !Jobs

(= Directory exists)

Inquiry: ^0\$RDIFFSDISK\Jobis

Response: ^0\$DI1 00

(= this directory or this data file does not exist)

Readout all filenames of a directory

Inquiry: ^0\$RDFFSDISK\Jobs*

Response: ^0\$DI1 06 COUNTER.JOB DATE.JOB DM.JOB Testprint.job
 LINIE.job LJ24 Test.job

Display all ISO Fonts in the fonts directory

Inquiry: ^0\$RDFFSDISK\Fonts\ISO*.lft

Response: ^0\$DI1 09 ISO1_5x5.lft ISO1_7x5.lft ISO1_9x7.lft ISO1_12x8.lft
 ISO1_14x10.lft ISO1_16x10.lft ISO1_24x18.lft ISO1_32x20.lft
 ISO1_5x7flue.lft

\$DI Directory-Info (Machine → Host)

The directory-info is transmitted in blocks and every block has max. 32 entries. The DirInfo contains

Par1: 0= further blocks follow, 1=last block

Par2: Number of entries in this block

Par3: Entry1

Par4:

Each single entry has a data filename.

The directory name is preceded by a !

\$DF Delete file (Drucker ← Host)

Deletes the specified file on the printer.

Parameter 1: File with path



Attention!

Inadvertent deletion of important files may result in further errors. Print jobs may not be available anymore!
 It is only possible to delete files from the default directory.

\$RF Request File (Host → Machine)

Sends a FileTransfer inquiry to the machine.

Par1: Data file with directory information

To copy a job of the internal Flashdisk to the host, you have to send:

^0\$RFFFSDISK\Jobs\MyJob1.job<CR>

The machine responds with FH and FT-instructions.

\$FH File Header (Host ↔ Machine)

File-transfers are implemented from host to machine as well as from machine to host. Each FileTransfer has to be started with a Header. It contains:

Par1: FileName with path. As path is always transferred the path on the machine side.

Par2: Number of the following FT-blocks. In case a 0 is transferred, the required data file cannot be opened.

The header is not acknowledged.

The printer starts a file transfer as answer to the command \$RF Request File (Host → Machine).

The host is able to start a file transfer directly with the file transfer header.

Hints:

The printer is not able to deliver empty files. The answer of the printer does not differ whether the files are empty or non-existent. But it is possible to create an empty file on the printer by sending a data block without content.

\$FT File Transfer (Host ↔ Machine)

With this instruction you can transfer a File Transfer block.

Par1: Block number

Par2: Block check (=Checksum modulo 256 of all data bytes of this block)

Par3: Data. Max. 2048 bytes.

The data are transmitted as nibble-string with 4 bit per character. The first nibble has the bits 7-4 of the first byte, the second the bits 3-0, the third nibble the bits 7-4 of the second byte etc.. The nibbles are coded as values from A...P (A=0, P=15).

After the receiver has analyzed the package, a **\$FA File transfer-acknowledge (Printer ↔ Host)** has to be returned:

\$FA File transfer-acknowledge (Printer ↔ Host)

Acknowledge for received package

Parameter 1: Block number

Parameter 2: 0=transfer error, block has to be resubmitted
1=understood, block is OK
2=Cancel file transfer completely

Annex A Transmissions with CRC32

To guarantee a secure data transfer also with a RS232 transmission and interfering field environment, the protocol offers a CRC32 procedure.

This support does not apply to the file transfers (for this separated block check numbers are used. See \$FT File Transfer)

If a transmission with CRC32 should be carried out is always initiated by the host prior to every transmission.

For the procedure of a CRC32 transmission the following commands and parameter transfers are important:

ParameterTransfer: NextCRC32 (Implementation Host<->LJ3)
 $\wedge 0=NR$ *Par1 = CRC32 Checksum*

ParameterTransfer: CRCFailed (Implementation Host<-LJ3)
 $\wedge 0=FC$ *Par1 = CRC32 Checksum*

Command:: CRCok (Implementation Host<-LJ3)
 $\wedge 0!OK$

Description of operation:

If you would like to have a secured transmission, the host calculates previous to every transmission the CRC32 checksum of the next transmission string and transfers it with $\wedge 0=NC$ to the printer.

It memorizes this CRC32 checksum, calculates and compares the CRC32 checksum of the transfered string for the next incoming communication string.

If both are unequally a response with the checksum which has been calculated in the LJ3 will be sent to the host $\wedge 0=FC$.

The host is then able e.g. to carry out a transmission repetition.

If the checksum is okay the printer responds (prior to the transmission of possible data) with $\wedge 0!OK$.

If an inquiry of data ($\wedge 0?XX$) with CRC32 has been sent to the printer a CRC32 checksum is transfered from the printer to the host prior to every response line.

Example of a (Hyperterminal) sequence with CRC32:

Query of the current used jobs (H = Host, L = printer):

$\wedge 0=NR3957421711$	➔ CRC32 of the following $\wedge 0?JL$ inquiry (H->L)
$\wedge 0?JL$	➔ inquiry itself (H->L)
$\wedge 0!OK$	➔ the confirmation of the LJ3 the CRC OK (H<-L)
$\wedge 0=NR3560773416$	➔ CRC32 of the pending response (H<-L)
$\wedge 0=JL\backslash FFSDISK\backslash JOBS\backslash Testprint.job$	➔ response itself (H<-L)

Annex B Mailing (Database) operation

The commands and parameter transfers which are relevant for the mailing operation are:

^0=MR String

^0=CM Par1

^0?SM and accordant response from LJ3 ^0=SM

^0!FF

Info regarding performance/Procedure:

- For a print stop all FIFO contents and parameters are deleted (**Stop at record number** = 0)
 Texts which are transmitted by the protocol can be changed in length. That means the „Wildcards“, which are set in the editor are only used for the editor after rebooting of the job or by sending a “Zero” string.
 If a fixed text length is required this has to done in the application which sends the mailing text. (is also valid for “Extern Text”)
- In the printer all mail messages are checked for ascending numbering during readout of the FIFO and in case this regulation can be not observed a „Warning“ message is generated. Exceptions are mail records with 0.
- If the **Stop at record number** value is >0, the print will be automatically stopped by reaching this number.
- The printer checks the numbering with each PrintGo(PG). With RecordNo 0 no check will be done and when the mailing buffer gets empty, with each PG the last received record will be printed once more. When the record numbers are numbered, this numbering will be checked. When this is not correct, a stop print and an error will be caused. When there is a numbering, the mailing buffer may not get empty, otherwise an error and a Stop print will be caused.
- Around 3 Milliseconds before a PG is coming, the mailing buffer must have at least one record. Is the speed 180 m/min or more, the mailing buffer must have at least one record 3 Milliseconds before the last stroke is printed.

Due to the above described performance the following is applied for the consecutive record number:

1. It has to be always available
2. It has to be 0 or numbered in ascending order and serially (No = No+1)
3. The first number of a new number sequence has to be always 0.

Example: 0 0 0 0 500 501 502 503 0 100 101 102 ...

Hint:



If Extern Text or Mailing operation is carried out with Unicode Fonts (defined in the accordant job) the text data has to be transferred hexadecimal coded!

e.g. "Hello World" has to be transferred as following
 00480065006C006C006F00200057006F0072006C0064

Further information regarding the conversion of Unicode can be found in the documentation of the script language.

Complete example about how to carry out communication:

A print job was created at the printer which contains a mailing object. The printer was started and the print job was loaded. The printer is running.
 The content of a database with 100.000 records shall be printed. Each record contains a last name. 22.117 records are already printed. Today the printout shall be continued seamlessly.

Direction of communication: PC => Printer (Blue)
 Printer => PC (green)

Status query	^0?RS
Status answer The printer is ready and there are no errors pending.	^0=RS2 5 0 0 9 1
Mailing Status query	^0?SM
Status answer The buffer memory is empty and has space for 256 entries.	^0=SM256 0 0 0 1
Setting the number of the record at which the printout shall stop. This is in order to avoid a error message when the last record is reached.	^0=CM100000
Status query to check whether the number of the last record to be printed was transferred correctly.	^0?SM

Status answer The printer has received and stored the number of the last record to be printed correctly.	<code>^0=SM256 0 0 100000 1</code>
Send the first mailing record. The buffer is always filled to the maximum which is 256 entries. Today, the first entry from data base is 22.118. Therefor this number is the first to be transferred to the printer. The consecutive numbering grants that the printer do not miss or double print a record. In order to reach an optimal transfer speed all entries are send in one block!	<code>^0=MR22118 Maier</code> <code>^0=MR22119 Mailänder</code> <code>^0=MR22120 Malasse</code> <code>^0=MR22121 Mann</code> <code>^0=MR22122 Marahrenz</code> <code>^0=MR22123 Marjenau</code> <code>^0=MR22124 Marks</code> <code>^0=MR22125 Marmelstein</code> <code>^0=MR22126 Marohn</code> ... <code>^0=MR22370 Michel</code> <code>^0=MR22371 Mickenhagen</code> <code>^0=MR22372 Middelberg</code> <code>^0=MR22373 Middelbüscher</code>
Status query in order to check whether all records were transferred.	<code>^0?SM</code>
Status answer The printer has received all records and the first record was already loaded for printing. For this reason the buffer memory has only 255 entries!	<code>^0=SM256 255 0 100000 1</code>
Print start	<code>^0!GO</code>
Status query	<code>^0?RS</code>
Status answer The printer has started to print the entries.	<code>^0=RS2 6 0 0 9 0</code>
Mailing Status query	<code>^0?SM</code>
Status answer. The first two records were already printed.	<code>^0=SM256 253 22119 100000 1</code>
... From now on the two status queries will be repeated every second. The further records will be sent in blocks by 50 records in order to keep the load for the communication process at a low level. In case an error occurs the print out will be stopped. After 50 records were printed the next block with 50 records shall be transferred. ...	
Status query	<code>^0?RS</code>
Status answer The printer started the printout.	<code>^0=RS2 6 0 0 9 0</code>
Mailing Status query	<code>^0?SM</code>
Status answer	<code>^0=SM256 198 22175 100000 1</code>

Transfer of the next 50 records.	<code>^0=MR22374 Middeldorf</code> <code>^0=MR22375 Middeldorff</code> <code>^0=MR22376 Middendarp</code> <code>^0=MR22377 Middendorf</code> <code>^0=MR22378 Middendorp</code> <code>^0=MR22379 Milges</code> ... <code>^0=MR22422 Moritzer</code> <code>^0=MR22423 Morris</code>
Status query	<code>^0?RS</code>
Status answer	<code>^0=RS2 6 0 0 9 0</code>
Mailing Status query	<code>^0?SM</code>
Status answer During the transfer of the records there were already further records printed.	<code>^0=SM256 249 22219 100000 1</code>
... From now on the two status queries will be repeated every second. As soon as the buffer has space for the next 50 entries the next block will be sent. This is repeated until all records from the data base will be transferred. After the last block was transferred the buffer memory will empty step by step.. ...	
Transfer of the last record	<code>^0=MR99997 Zimmerman</code> <code>^0= MR99998 Zinser</code> <code>^0= MR99999 Zupke</code> <code>^0= MR100000 Zwick</code>
Mailing status query	<code>^0?SM</code>
Status answer	<code>^0=SM256 249 99750 100000 1</code>
...	
Mailing status query	<code>^0?SM</code>
Status answer	<code>^0=SM256 127 99872 100000 1</code>
...	
Mailing status query	<code>^0?SM</code>
Status answer	<code>^0=SM256 49 99950 100000 1</code>
...	
Mailing status query	<code>^0?SM</code>
Status answer	<code>^0=SM256 3 99996 100000 1</code>
After the last printout the printer stopps automatically and displays a message. This can also be checked in the status answers.:	
Status query	<code>^0?RS</code>
Status answer Printout finished. Message 1223 is pending. („Last data base entry was printed“).	<code>^0=RS2 5 -1711274809 0 9</code> <code>0</code>
Mailing status query	<code>^0?SM</code>
Status answer The number last printed is the last	<code>^0=SM256 0 100000 0 1</code>

number from the data base.

Comparison of data transfer modes

There are three different methods to print variable texts:

1. Load a new print job.
2. Send an Extern Text
3. Send a Mailing Record

Die drei Varianten haben ihre Stärken und Schwächen:

Print Job	Extern Text	Mailing Record
Flexible. Position, Font, etc. are adjustable.	Print job is fix. Only the text can be changed	
Graphics are possible		
40..150 ms for loading a print job depending on the kind of job. Only for slow applications.	40..150 ms for loading a print job depending on the text. Suitable for fast applications..	The memory buffer provides a print job loading without timing restrictions. Maximum speed is possible.
Complex. The structure of the job script and the communication protocol must to be known.	Simple. Only text is transferred.	Complex. The structure of the communication protocol must to be known. Status messages have to be considered.
It is not possible to avoid double prints.	Double prints are possible but they may cause an error.	Double prints are always inhibited.

History

V1.3	Command !LN added
V1.4	Command =FL added
V1.5	Command =RC added
V1.6	Command !EM added
	Command =PR added
V1.7	Command ?CC: Total print counter added
V1.7.1	Command =AC added
V1.7.2	Parameter 9 in command =OC added
V1.7.3	New: Command ?ET
V1.7.4	Command PR: Speed at internal encoder pulses is now dm/min
V1.7.5	Command =MR: Graphic Mailing added (since V60.0.1.22)
V1.7.6	Command =MR: Property-Parameter added (since V60.0.1.22)
V1.7.7	Command =CO implemented (since V60.0.25.0)
V1.8.0	Commands ?DS, =DS, =SW und \$DF added.
V1.8.1	Command ?AC added, ?JC und =JC removed. Extended details for some commands.
V1.8.2	Extended details for some commands. Information regarding special characters and extendable parameter list added. New parameter „Job change“ for =RS.
V1.9.0	Documentation about restrictions for the JET2neo added.
V1.9.1	Clarifications about the range of character sets and the handling of special characters added.
V1.9.2	Command =DL removed
V1.9.3	Empty commands (more 0x0d in succession) are allowed