SPPL

SAVEMA THERMAL TRANSFER PROGRAMMING LANGUAGE

(Revision-7)

Markalama ve Kodlama Makinaları San.Tic. Ltd. Sti.

Fevzi Cakmak Mah. Ahmet Petekci Cad. No :5L/1

Karatay / KONYA / TURKIYE

Tel : +90 332 239 2339

Fax : +90 332 239 2319

Web: www.savema.com.tr

Revision Notes

The following changes have been made to this document.

1- Modification Commands seperator is changed as **~gt~** (> character was used as separator). This changing is applied to SPMCTV, SPMCBV, SPMC2D, SPMCCV and SPMCSV commands.Look at Modification commands. (Page 76-79)
2- Create Database File command(SPLCDF) seperator is changed as **~gt~** (> character was used as separator).
   Column seperator is changed for datas which is used as parameter of SPLCDF command. **~sc~** text is used as column seperator for each row of data(; character was used as separator).Look at SPLCDF command explanation. (Page 70)
3- SPLCDB (Clear Data Buffer) command is added. (Page 73)
4- SPLGFN (Get Field Names) command is added. (Page 74)
5- SPPSTA (Status of printer) command is added. (Page 82)
6- SPGGCP (Get current print count) command is added. (Page 84)

Note: Controller software version must be minimum v3.13.5 to use above changes.

CONTENTS

1.General Rules for SPPL

SPPL is programming language for controlling Savema Thermal Transfer printer over Ethernet/RS-232 communication.

1.1 SPPL Command Structure

SPPL commands have some rules which is shown in below;

- SPPL commands starts with **~** character and ends with **^** character.(ie *~SPPSAP^*)
- SPPL commands seperates with **|** character for more than one commands. (ie *~SPPSLQ{1000}| SPPSAP^*). This character only have to be used for seperate commands.
- SPPL command parameters defines between **{** and **}** characters.
- (ie. *~ SPPSLQ {1000}^*)
- Set and Change Commands parameters seperates with **>** character. (ie. *~SPCSSC{115200>None>8>1}^)*
  Get Commands parameters seperates with **<** characters. (ie. *~ SPGRES{ SPCGSC:115200<None<8<1}^)*
  Modification commands and Create Data File command parameters seperates with **~gt~** text. (ie. *~SPMCSV{text1~gt~Savema}^*
- SPPL command letters are created according to some rules.
  - SP means Savema Printer
  - 3rd character indicates Command type. Command types with letters are;
    - C : Configuration Commands
    - L : Label Commands
    - M : Modification Commands
    - P : Print Commands
    - G : General Commands
  - Last 3 characters are abbreviation of Command Name

Forexample; SPCSNC command letters are separating like below;

- SP is Savema Printer
- C is Configuration Command type
- SNC is Set Network Configuration

1.2 Command List

SPPL commands seperates according to using type. There are 5 types of command groups in SPPL.

- **Configuration Commands**
- **Label Designing Commands**
- **Modification Commands**

- **Print Commands**
- **General Commands**
- **Traverse Commands**

Note : Some commands doesn't supported by all printers. If so, printer sends FAIL message when getting incompatible command. Please see part 9.1) Command Limitations .

Note : Data transfer time changes according to communication type and speed. If data is big, transfer time increases.

All of commands are listed in below as a table format

SAVEMA PRINTER PROGRAMMING LANGUAGE COMMAND LIST			
Command	**Explanation**	**Usage**	**Example**
CONFIGURATION COMMANDS			
SPCSDT	Set System Date&Time and Time Offset	**~SPCSDT{DD>MM>YYYY>HH>mm>SS>OO}^**	~SPCSDT{25>01>2015>11>36>00>00}^
SPCGDT	Get System Date&Time and Time Offset	**~SPCGDT^**	~SPCGDT^
SPCSNC	Set Network Configuration	**~SPCSNC{IP Address>Subnet Mask>Gateway>Port number}^**	~SPCSNC{192.168.1.123> 255.255.255.0>192.168.1.1>9100}^
SPCGNC	Get Network Configuration	**~SPCGNC^**	~SPCGNC^
SPCSSC	Set RS-232 Configuration	**~SPCSSC{Baud Rate> Parity> Data Bits> Stop Bits}^**	~SPCSSC{115200>None>8>1}^
SPCGSC	Get RS-232 Configuration	**~SPCGSC^**	~SPCGSC^
SPCSPS	Set Print Speed (Intermitttent)	**~SPCSPS{Print Speed}^**	~SPCSPS{200}^
SPCGPS	Get Print Speed (Intermitttent)	**~SPCGPS^**	~SPCGPS^
SPCSPD	Set Print Delay value	**~SPCSPD{Print Delay}^**	~SPCSPD{10}^
SPCGPD	Get Print Delay value	**~SPCGPD^**	~SPCGPD^
SPCSDV	Set Darkness(Contrast) Value	**~SPCSDV{Contrast}^**	~SPCSDV{100}^
SPCGDV	Get Darkness(Contrast) Value	**~SPCGDV^**	~SPCGDV^
SPCSPR	Set Print Rotation	**~SPCSPR{Print Rotation}^**	~SPCSPR{180}^
SPCGPR	Get Print Rotation	**~SPCGPR^**	~SPCGPR^
SPCSHP	Set Horizontal Position	**~SPCSHP{Horizontal Position Value}^**	~SPCSHP{0}^
SPCGHP	Get Horizontal Position	**~SPCGHP^**	~SPCGHP^
SPCSVP	Set Vertical Position	**Will be used in then future**	
SPCGVP	Get Vertical Position	**Will be used in then future**	

SPCSMO	Set Mirroring Option	**~SPCSMO{Mirroring Option}^**	~SPCSMO{0}^
SPCGMO	Get Mirroring Option	**~SPCGMO^**	~SPCGMO^
SPCSRS	Set RibbonSave Mode	**~SPCSRS{Direction>Column No>Shifting length }^**	~SPCSRS{0>2>4}^
SPCGRS	Get RibbonSave Mode	**~SPCGRS^**	~SPCGRS^
SPCSIC	Set Internal Contact Mode (Continuous only)	**~SPCSIC{Internal Contact Mode State> Package length }^**	~SPCSIC{1>100}^
SPCGIC	Get Internal Contact Mode (Continuous only)	**~SPCGIC^**	~SPCGIC^
SPCSTC	Set Trigger Contact Mode (Continuous only)	**~SPCSTC{Trigger Contact Mode State> Print Count>Package length }^**	~SPCSTC{1>3>100}^
SPCGTC	Get Trigger Contact Mode (Continuous only)	**~SPCGTC^**	~SPCGTC^
SPCSAS	Set All Settings	**~SPCSAS{Print Speed>Print Delay>Darkness Value>RibbonSave Mode Direction> RibbonSave Mode Column No> RibbonSave Mode Package Length>Internal Contact Mode State>Internal Contact Mode Package Length>Trigger Contact Mode State>Trigger Contact Mode Print Count >Trigger Contact Mode Package Length}^**	~SPCSAS{300>2>100>0>1>0>0>30>1>3>60 }^
SPCGAS	Get All Settings	**~SPCGAS^**	~SPCGAS^
SPCSSP	Set System Parameter	**~SPCSSP{Parameter No> Parameter value}^**	~SPCSSP{1>25}^
SPCGSP	Get System Parameter	**~SPCGSP{Parameter No }^**	~SPCGSP{1}^
SPCSPA	Set All System Parameters	**~SPCSPA{P1>P2>P3,P4>P5>P6>P7>P8>P9>P 10>P11>P12>P13>P14>P15>P16>P17>P18> P19>P20}^** **P means Parameter**	~SPCSPA{25>27>300>200>31>77>0>24>25 >0>12>65>0>5>0>23>0>4>0>0>400}^
SPCGPA	Get All System Parameters	**~SPCGPA^**	~SPCGPA^
SPCSSL	Set System Language	**~SPCSSL{System Languge Code }^**	~SPCSSL{02}^
SPCGSL	Get System Language	**~SPCGSL^**	~SPCGSL^
SPCSAP	Set Administrator Password	**~SPCSAP{System Password}^**	~SPCSAP{123456}^
SPCGAP	Get Administrator Password	**~SPCGAP^**	~SPCGAP^
SPCSFS	Return to Factory Settings	**~SPCSFS^**	~SPCSFS^
SPCSPM	Set Print Request Message	**~SPCSPM{Print Request Active|Passive> Print Message}^**	~SPCSPM{0>OK}^
SPCGPM	Get Print Request Message	**~SPCGPM^**	~SPCGPM^

LABEL DESIGNING COMMANDS			
SPLTDS	Create Template Datas and Template Structure	**~SPLTDS{Template Datas}^**	~SPLTDS{<Template> <General> <MachineType>53x70l</MachineType> <Name>temp1_53.ronx</Name> <Width>640</Width> <Height>480</Height> </General> <Object> <ObjectType>Text</ObjectType> <Name>text1</Name> <X>10</X> <Y>63</Y> <W>105</W> <H>33</H> <Rotate>180</Rotate> <Content> <Data>savema Printer</Data> <Source>Internal</Source> <MagnificationRatio>100</ MagnificationRatio> </Content> <Font> <Name>Arial</Name> <Size>15</Size> <Style>Bold,Italic</Style> </Font></Object> </Template>}^
SPLLTF	Load Template from Printer	**~SPLLTF{Template File Name}^**	~SPLLTF{temp1_53.ronx}^
SPLGAT	Get Active Template	**~SPLGAT^**	~SPLGAT^
SPLGST	Get Stored Templates	**~SPLGST^**	~SPLGST^
SPLGSD	Get Stored Data Files	**~SPLGSD^**	~SPLGSD^
SPLCDF	Create Data File	**~SPLCDF{Data File Name~gt~File Content}^**	~SPLCDF{sample.csv~gt~abc bce cde}^
SPLDTF	Delete Template	**~SPLDTF{Template File Name}^**	~SPLDTF{temp1_53.ronx}^
SPLDTA	Delete All Template	**~SPLDTA^**	~SPLDTA^
SPLDDF	Delete Data File	**~SPLDDF{Data File Name}^**	~SPLDDF{datafile1.csv}^
SPLDDA	Delete All Data File	**~SPLDDA^**	~SPLDDA^
SPLCDB	Clear Data Buffer	**~SPLCDB^**	~SPLCDB^
SPLGFN	Get Field Names	**~SPLGFN{ Template File Name }^**	~SPLGFN{temp1_53.rox}^
MODIFICATION COMMANDS			
SPMCTV	Changing Text Value	**~SPMCTV{Name of Object~gt~Text Value}^**	~SPMCTV{brand_txt**~gt~**SAVEMA}^
SPMCBV	Changing Barcode Value	**~SPMCBV{Name of Object ~gt~Barcode Value}^**	~SPMCBV{barcodeno**~gt~**8691234567890}^
SPMC2D	Changing 2D Barcode Value	**~SPMC2D{ Name of Object ~gt~Barcode Value}^**	~SPMC2D{qrcodeno**~gt~**savema12345}^
SPMCCV	Changing Counter Value	**~SPMCCV{Name of Object ~gt~Counter Value}^**	~SPMCCV{counter1**~gt~**000055}^

SPMCSV	Changing Selected Values	**~SPMCSV{Name of Object~gt~Text Value~gt~Name of Object~gt~Text Value}^**	~SPMCSV{ brand_txt**~gt~**SAVEMA**~gt~** qrcodeno**~gt~**savema12345}^
PRINT COMMANDS			
SPPSAP	Start Automatically Print	**~SPPSAP^**	~SPPSAP^
SPPSLQ	Set Print Count for Limited print	**~SPPSLQ{Print Quantity}^**	~SPPSLQ{1000}^
SPPGLQ	Get Print Count for Limited print	**~SPCGLQ^**	~SPPGLQ^
SPPSTP	Stop Print	**~SPPSTP^**	~SPPSTP^
SPPOTP	One Test Print	**~SPPOTP^**	~SPPOTP^
SPPSTA	Status of Printer	**~SPPSTA^**	~SPPSTA^
GENERAL COMMANDS			
SPGSUM	Send User Message to Printer	**~SPGSUM{User Message}^**	~SPGSUM{Package finished. Please stop printer}^
SPGRES	General Response Command From Printer	**~SPGRES{Response}^**	~SPGRES{950225}^
SPGGTP	Get Total Print Count	**~SPGGTP^**	~SPGGTP^
SPGGFW	Get Firmware Version	**~SPGGFV^**	~SPGGFV^
SPGGRR	Get Remaining Ribbon (From Cassette models)	**~SPGGRR^**	~SPGGRR^
SPGGSN	Get Serial Number of Printer	**~SPGGSN^**	~SPGGSN^
SPGGCP	Get Current Print Count	**~SPGGCP^**	~SPGGCP^
TRAVERSE COMMANDS			
SPTSPS	Set Pack Size	**~SPTSPS{Pack Size(mm)}^**	~SPTSPS{60}^
SPTGPS	Get Pack Size	**~ SPTGPS^**	~SPTGPS^
SPTSPC	Set Print Count	**~ SPTSPC{Print Count}^**	~SPTSPC{5}^
SPTGPC	Get Print Count	**~SPTGPC^**	~SPTGPC^
SPTSPP	Set Print Position	**~SPTSPP{Print position(mm)}^**	~SPTSPP{10}^
SPTGPP	Get Print position	**~SPTGPP^**	~SPTGPP^
SPTSPD	Set Pack Distance From Beginning	**~SPTSPD{Pack distance(mm)}^**	~SPTSPD{50}^
SPTGPD	Get Pack Distance From Beginning	**~ SPTGPD^**	~SPTGPD^
SPTSPA	Set Printing Area	**~ SPTSPA{Printing Area}^**	~SPTSPA{400}^
SPTGPA	Get Printing Area	**~SPTGPA^**	~SPTGPA^
SPTSTP	Set All Traverse Parameters	**~SPTSTP{Pack Size>Print Count>Print Position>Pack Distance>Printing Area}^**	~SPTSTP{60>5>10>50>400}^
SPTGTP	Get All Traverse Parameters	**~SPTGTP^**	~SPTGTP^

Table-1) Command List

2. Configuration Commands

Configuration Commands allows to make changing on printer settings. Some settings affects printer working. So, must be carefull using this commands

2.1 Set/Get System Date&Time and Time Offset

SPCSDT :Allows to adjust system date and time and Time Offset.

Time offset is using for changing template date before or after midnight. This is changing between -12(before midnight) and 12(after midnight).

Date and time value and time offset value must be sent as a parameter with this command.

Printer sends OK message when setting date&time is successed or sends FAIL message when setting date&time is failed.

Using	**~SPCSDT{DD>MM>YYYY>HH>mm>SS>OO}^** **Parameters;** **DD:** Day (2 digits). Day Value must be between 00-31. February has 28-29 days and some months has 30 days. So, be carefull when setting the value of the day. **MM :** Month (2 digits). Month Value must be between 00-12. **YYYY:** Year (4 digits). Year value must be between 1900-3000. **HH :**Hour(2 digits). Hour value is adjusted according to 24 Hours. So, hour value must be between 00-23. **mm :** Minute(2 digits). Minute value must be between 00-59. **SS :** Second(2 digits). Second value must be between 00-59. **OO :** Time Offset(2 digits). Time offset value must be between -12 and 12.
Example	~SPCSDT{25>07>2017>11>36>00>00}^ – with timeoffset ~SPCSDT{25>07>2017>11>36>00>02}^ – without timeoffset Return Value(On Successed) : ~ SPGRES{ SPCSDT:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSDT:FAIL}^

SPCGDT : Returns system date&time and time offset value from printer. İf any problem happens while commands processing, printer sends FAIL message.

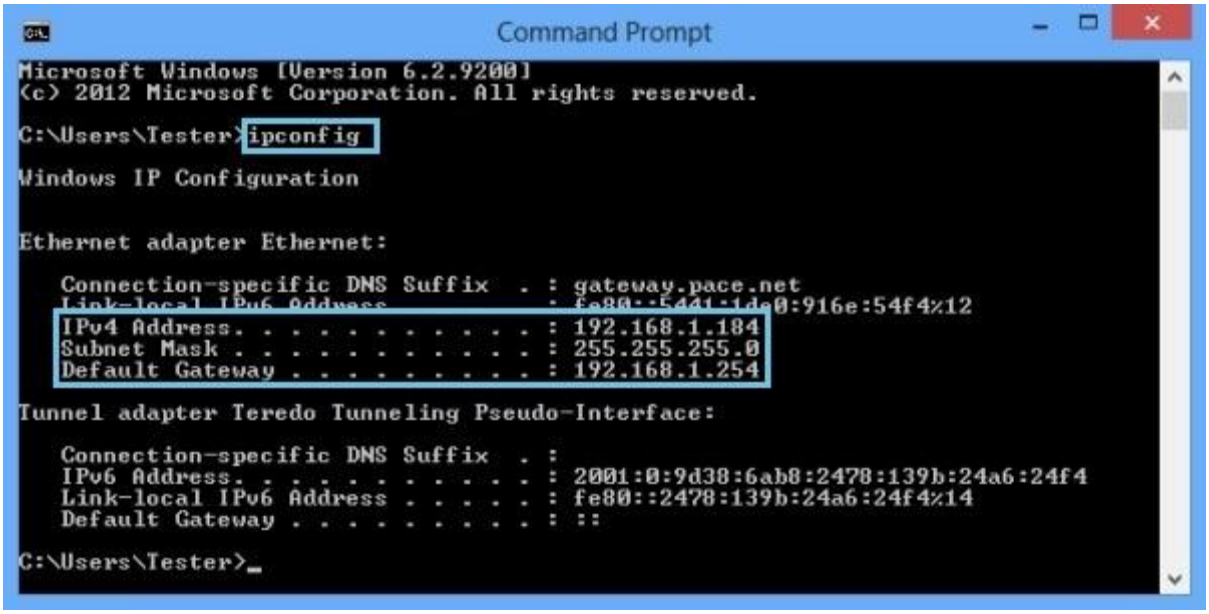
Using	**~SPCGDT^**
Example	~SPCGDT^ Return Value(On Successed) : ~ SPGRES{ SPCGDT :25<07<2017<11<36<00<00}^

2.2 Set/Get Network Configuration

SPCSNC : Allows to configure network parameters of printer. This parameters adjust via Ethernet communication, existing communication will be disconnected after finish adjustment and needs to connect with new parameters(IP address …etc). If RS-232 is used for configure, there is no disconnection problem after adjust network parameters. SPCSNC commands is sending with some parameters for configuration. Printer sends OK message when setting network configuration is successed or sends FAIL message when setting network configuration is failed.

Using	**~SPCSNC{IP Address>Subnet Mask>Gateway>Port number}^** **Parameters;** **IP Address:** Printer IP(Internet Protocol) Address must be IPv4 standart This IP address must be unique in network. Otherwise devices(have same IP address) can be conflict while working in same network. **Subnet Mask :**Subnet Mask must be adjust according to IP Address class. **Gateway:**This addressis same one network and all devices(same network) uses same gateway address. Gateway address have to be same for communicate with printer. **Port Number :** This is 9100 in printers. So, must be use 9100 as a port number. Return Value(On Successed) : ~ SPGRES{ SPCSNC:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSNC:FAIL}^
Example	~SPCSNC{192.168.1.123>255.255.255.0>192.168.1.1>9100}^ ~SPCSNC{192.168.1.100>255.255.255.0>192.168.1.1>9100}^

Note : Network parameters(Subnet Mask, Gateway Address) can be learn from Command Prompt with **ipconfig** command like below image.



SPCGNC : Returns network parameters.

Using	**~SPCGNC^**
Example	~SPCGNC^ Return Value(On Successed) : ~ SPGRES{ SPCGNC:192.168.1.123<255.255.255.0<192.168.1.1<9100}^

2.3 Set/Get RS-232(Serial) Configuration

SPCSSC : Allows to set RS-232 parameters in printer. Please use 9 pins(DB9) standart **crossover** cable for communicate with printer via RS-232. RS -232 is configuring via RS-232 or Ethernet. İf use RS-232 for configuration, new parameters must be apply to device which will comunicate with printer.
Printer sends OK message when setting RS-232 configuration is successed or sends FAIL message when setting RS-232 configuration is failed.

Using	**~SPCSSC{Baud Rate>Parity>Data Bits>Stop Bits}^** **Parameters;** **Baud Rate:**Adjusts data flowing speed as bits per second. İt must be happen between 1200 – 115200bps. Can be 1200-2400-4800-9600-14400-19200-28800-38400-56000-57600-115200. Printer is using 115200 bps for communication as a default but it can change. **Parity :**Can be None-Odd-Even-Mark-Space. Printer is using None value for

	parity as a default but it can change. **Data Bits:**Can be 5-6-7-8. Printer is using 8 bits for data bits as a default but it can change. **Stop Bits :**Can be 1-1.5-2. Printer is using 1 bit for stop bits as a default but it can change. Return Value(On Successed) : ~ SPGRES{ SPCSSC:OK}^ Return Value(On Failed) : ~ SPGRES{SPCSSC:FAIL}^
Example	~SPCSSC{115200>None>8>1}^

SPCGSC: Returns RS-232(Serial) parameters of printer.

Using	**~SPCGSC^**
Example	~SPCGSC^ Return Value(On Successed) : ~ SPGRES{ SPCGSC:115200<None<8<1}^

2.4 Set/Get Print Speed(Intermitttent only)

SPCSPS: Allows to set print speed(mm/sec) of printer. Printers speed is adjusting for intermittent models because continuous type printers gets print speed from encoder and encoder gets speed from media(package,label…etc) flowing speed.
Printer sends OK message when setting print speed is successed or sends FAIL message when setting print speed is failed.

Using	**~SPCSPS{Print Speed}^** **Parameters;** **Print Speed:**Specified according to millimeter per second. Print speed value must be between 150-400. **Note :** If user want to print on hard meda or use resin type ribbon, print speed must be maximum 200. Please look at Service manual of related printer for more information. Return Value(On Successed) : ~ SPGRES{ SPCSPS:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSPS:FAIL}^
Example	~SPCSPS{200}^

SPCGPS : Returns print speed of printer..

Using	**~SPCGPS^**
Example	~SPCGPS^ Return Value(On Successed) : ~ SPGRES{ SPCGPS:200}^

2.5 Set/Get Print Delay value

SPCSPD : Allows to set print delay after print signal arrives. Specified in millimeter for continous models and specified in milliseconds for intermittent models.
Printer sends OK message when setting print delay is successed or sends FAIL message when setting print delay is failed.
This value changes after printer starts to print.

Using	**~SPCSPD{Print Delay}^** **Parameters;** **Print Delay:**Print Delay is start of print time after print signal arrives.Specified in millimeter(continuous) or millisecond(intermittent). Return Value(On Successed) : ~ SPGRES{ SPCSPD:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSPD:FAIL}^
Example	~SPCSPD{10}^

SPCGPD : Returns print delay of printer.

Using	**~SPCGPD^**
Example	~SPCGPD^ Return Value(On Successed) : ~ SPGRES{ SPCGPD:0}^

2.6 Set/Get Darkness(Contrast) Value

SPCSDV : Allows to set print darkness(contrast) of printer. Darkness can change according to media type. Can be increased when print quality is not good on media and also can increase ribbon type.
Printer sends OK message when setting darkness is successed or sends FAIL message when setting darkness is failed.
This value changes after printer starts to print.

Using	**~SPCSDV{Contrast}^** **Parameters;** **Contrast:** Darkness(Contrast) value must be between 60-120. Return Value(On Successed) : ~ SPGRES{PCSDV:OK}^ Return Value(On Failed) : ~ SPGRES{PCSDV:FAIL}^
Example	~SPCSDV{100}^

SPCGDV : Returns darkness(contrast) value of printer.

Using	**~SPCGDV^**
Example	~SPCGDV^ Return Value(On Successed) : ~ SPGRES{ SPCGDV:100}^

2.7 Set/Get Print Rotation

SPCSPR : Allows to adjust print rotation of template.
Printer sends OK message when setting print rotation is successed or sends FAIL message when setting print rotation is failed.

Using	**~SPCSPR{Print Rotation}^** **Parameters;** **Print Rotation:**Can be 0-90-180-270. 0 : Print direction is normal 90 :Print directionis 90 degrees of clockwise 180 : Print direction is reverse

	270 : Print directionis 270 degrees od clockwise or 90 degrees of anti-clockwise **Note :** Template height cannot be higher than printhead size while using 90 or 270 print direction or vice versa. For 32 mm models, template height must be maximum 32mm. For 53 mm models, template height must be maximum 53mm. For 107 mm models, template height must be maximum 107mm For 107x75I model, width is higher than height (107 > 75), 90 or 270 rotation doesn't supported. Return Value(On Successed) : ~ SPGRES{ SPCSPR:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSPR:FAIL}^
Example	~SPCSPR{180}^

SPCGPR: Returns print rotation value template.

Using	**~SPCGPR^**
Example	~SPCGPR^ Return Value(On Successed) : ~ SPGRES{SPCGPR:180}^

2.8 Set/Get Horizontal Position

SPCSHP : Allows to set horizontal position of print. This command changes print location horizontally and moves print to right side. If print is moving to outside of print area, overflowing part printt left side.

Printer sends OK message when setting horizontal position is successed or sends FAIL message when setting horizontal position is failed.

This value changes after printer starts to print.

Using	**~SPCSHP{Horizontal Position Value}^** **Parameters;** **Horizontal Position Value:**Horiontal Position value must be start 0. Maximum value of position changes according to print head types. For 32 mm models, horizontal position can increase maximum 48. For 53 mm models, horizontal position can increase maximum 80. For 107 mm models, horizontal position can increase maximum 160.

	Return Value(On Successed) : ~ SPGRES{ SPCSHP:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSHP:FAIL}^
Example	~SPCSHP{0}^

SPCGHP : Returns horizontal position value of print.

Using	**~SPCGHP^**
Example	~SPCGHP^ Return Value(On Successed) : ~ SPGRES{SPCGHP:0}^

2.9 Set/Get Vertical Position(It will be used in the future)

SPCSVP(Not Used.)

SPCGVP(Not used)

2.10 Set/Get Mirroring Option

SPCSMO : Allows to print template mirrored.

Printer sends OK message when setting mirroring option is successed or sends FAIL message when setting mirroring option is failed.

Using	**~SPCSMO{Mirroring Option}^** **Parameters;** **Mirroring Option:** This paramter must be 0 or 1. **0**: Mirroring is passive **1**: Mirroring is active Return Value(On Successed) : ~ SPGRES{ SPCSMO:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSMO:FAIL}^
Example	~SPCSMO{0}^

SPCGMO : Returns mirroring active or passive of print.

Using	**~SPCGMO^**
Example	~SPCGMO^ Return Value(On Successed) : ~ SPGRES{ SPCGMO:0}^

2.11 Set/Get RibbonSave Mode

SPCSRS : Allows to print more than one columns on same vertical or horizontal position. There is two-type RibbonSave mode. These are;

Vertical : Vertical RibbonSave mode must be used when template widthsmaller than half of printhead size. Otherwise printer prints on of another print in same vertical position. Look at ribbonsave schema in below.

Horizontal : (For only Intermittent models) If template objects has vertical gaps, Horizontal RibbonSave mode reduces ribbon consumption.Colıumn no and Shifting lengh should be adjust according to between objects gaps. Otherwise printer prints on of another print after gaps in same horizontal position.Look at ribbonsave schema in below.

Please look at Service manual of related printer for more information.

Printer sends OK message when setting ribbonsave mode is successed or sends FAIL message when setting ribbonsave mode is failed.

Using	**~SPCSRS{Direction>Column No>Shifting length }^** **Parameters;** **Direction:** Should be 0 or 1 . 0 is Vertical RibbonSave mode, 1 is Horizontal RibbonSaveMode(This is only for intermittent printers). 0 is Default value. **Column No:**Provides to select print count on same vertical position. 1 is default. 1 mean only one print in same vertical position.Can increase according to print width. İf it is higher than half of printhead size, it must be 1. Forexample, if printhead size is 53 mm and print width is 10mm, Can increase upto 5. **Shifting length :**Specifies distance between two prints for more than one columns. Specified in millimeter. Parameter must adjust according to print width. Forexample, if printhead is 53 mm and print width 8 mm, shifing length must be 3 mm. Otherwise, prirnter prints upon another print.This value starts from 0, end value can be increased upto appropriate value.

	Return Value(On Successed) : ~ SPGRES{SPCSRS:OK}^ Return Value(On Failed) : ~ SPGRES{SPCSRS:FAIL}^
Example	~SPCSRS{0>1>0}^ - For 1 column- No RibbonSave ~SPCSRS{0>2>4}^ - For 2 columns and Columns distance is 4 mm

SPCGRS : Returns RibbonSave Mode parameters.

Using	**~SPCGRS^**
Example	~SPCGRS^ Return Value(On Successed) : ~ SPGRES{ SPCGRS:0<1<0}^

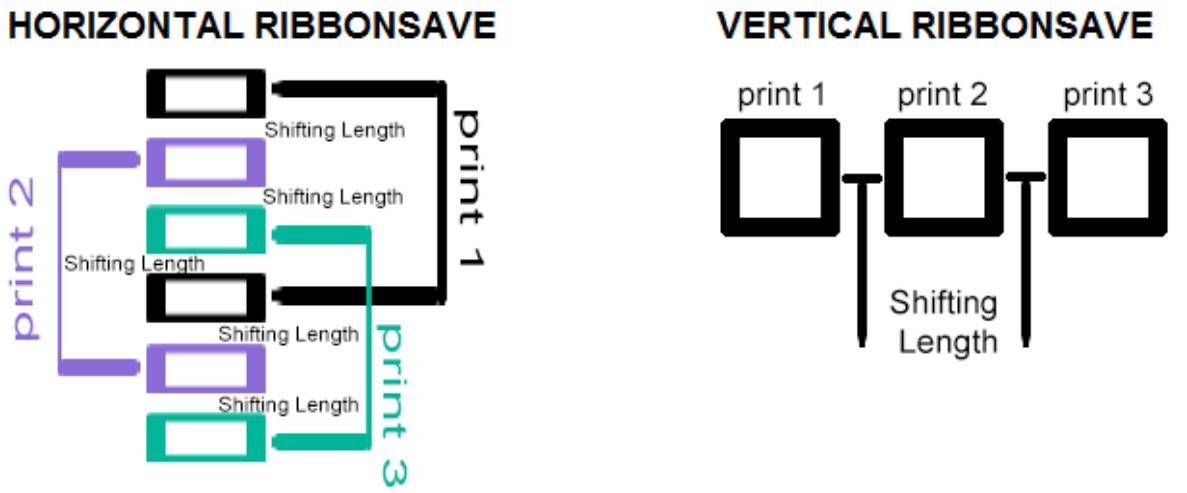


Figure : Exmaple scheme of Ribbonsave modes

2.12 Set/Get Internal Contact Mode(Continuous only)

SPCSIC : This command provides to printer prints without external print signal. This command is only using continuos printer and this type of printer prints at regular intervals without print signal (from photocell, pack machine..etc)

Printer sends OK message when setting internal contact mode is successed or sends FAIL message when setting internal contact mode is failed.

Using	**~SPCSIC{Internal Contact Mode State> Package length }^** **Parameters;** **Internal Contact Mode State:**To enable or disable Internal contact mode. This value must be 0 or 1. **0 :** Disable **1 :** Enable **Package length :**Specifies on package length for print. Printer prints at regular intervals according to package length. Specified in millimeter. This value must be between 35-1000 **Note:** Internal Contact mode doesn't work with Trigger Contact mode. So, if Internal Contact will be enabled, Trigger Contact Mode must be disabled. Return Value(On Successed) : ~ SPGRES{ SPCSIC:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSIC:FAIL}^
Example	~SPCSIC{1<100}^ - Printer prints per 100 mm without print signal ~SPCSIC{0<100}^ - Printer prints when external print signal comes

SPCGIC : Returns Internal Contact Mode parameters.

Using	**~SPCGIC^**
Example	~SPCGIC^ Return Value(On Successed) : ~ SPGRES{SPCGIC:1<200}^

2.13 Set/Get Trigger Contact Mode(Continuous only)

SPCSTC : This command provides to printer, prints more than one in one print signal. Printer sends OK message when setting trigger contact mode is successed or sends FAIL message when setting trigger contact mode is failed.

Using	**~SPCSTC{Trigger Contact Mode State>Print Count> Package length }^** **Parameters;** **Trigger Contact Mode State:**To enable or disable Trigger contact mode. This value must be 0 or 1. **0 :** Disable **1 :** Enable **Print Count :** Specifies how many print per contact. **Package lengh :**Specifies on package length for print. Printer prints at regular intervals according to package length. Specified in millimeter. This value must be between 35-1000 **Note:**Trigger Contact mode doesn't work with Internal Contact mode. So, if Trigger Contact will be enabled, Internal Contact Mode must be disabled. Because trigger contact mode Works with external print signal. Return Value(On Successed) : ~ SPGRES{ SPCSTC:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSTC:FAIL}^
Example	~SPCSTC{1>3>100}^ - Printer prints 3 times at 100 mm intervals after print signal comes ~SPCSTC{0>1>100}^ - Printer prints one time when external print signal comes

SPCGTC: Returns Trigger Contact Mode parameters.

Using	**~SPCGTC^**
Example	~SPCGTC^ Return Value(On Successed) : ~ SPGRES{SPCGTC:1<2<200}^

2.14 Set/Get All Settings

SPCSAS : Allows to set below settings.

1- Print Speed
2- Print Delay
3- Darkness Value
4- RibbonSave Mode Direction
5- RibbonSave Mode Column No
6- RibbonSave Mode Package Length
7- Internal Contact Mode State

8- Internal Contact Mode Package Length
9- Trigger Contact Mode State
10- Trigger Contact Mode Print Count
11- Trigger Contact Mode Package Length

Some settings are used according to printer type. (eg: Internal and Contact Mode is usable only Continuous models, Print Speed is usable in only Intermittent models.) 0 value can be used for unused parameters, All settings must be sent in proper sequence. Otherwise printer doesn't apply this settings.
Printer sends OK message when setting all settings are successed or sends FAIL message when setting all settings are failed.

Using	**~SPCSAS{Print Speed>Print Delay>Darkness Value>RibbonSave Mode Direction> RibbonSave Mode Column No> RibbonSave Mode Package Length>Internal Contact Mode State>Internal Contact Mode Package Length>Trigger Contact Mode State>Trigger Contact Mode Print Count >Trigger Contact Mode Package Length}^** **Note:** Above parameters explanation is showed with related settings command. So, please see related commands for more details. Forexample, for Print Speed look at SPCSPS(Set Print Speed) command expalanation. Return Value(On Successed) : ~ SPGRES{ SPCSAS:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSAS:FAIL}^
Example	~SPCSAS{300>2>100>0>1>0>0>30>1>3>60}^ - Set all system settings according to specified values.

SPCGPA : Returns all system settings value.

Using	**~SPCGAS^**
Example	~SPCGAS^ Return Value(On Successed) : ~ SPGRES{SPCGAS:300<2<100<0<1<0<0<30<1<3<60}^ -- All system parameters are returned with SPGRES command.

2.15 Set/Get System Parameter

SPCSSP : Allows to set selected system parameters. Savema Printers have 20 pieces of parameters and each parameter function is changing according to printer type. Please look at System Paramaters Explanation for more info at the end of this document.

Printer sends OK message when setting selected system parameter is successed or sends FAIL message when setting selected system parameter is failed.

Using	**~SPCSSP{Parameter No>Parameter value}^** **Parameters;** **Parameter No:**Specifies parameter number which will be changed. Value must be between 1 -20. **Parameter Value :**Specifies parameter value of selected system parameter. This parameter's minimum and maximum value is changing according to parameter function and printer type. Please look at System Paramaters Explanation for more info at the end of this document. Return Value(On Successed) : ~ SPGRES{ SPCSSP:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSSP:FAIL}^
Example	~SPCSSP{1>25}^ - Set First System Parameter to 25 ~SPCSSP{15>20}^ - Set 15th System Parameter to 20

SPCGSP : Returns selected system parameter value.

Using	**~SPCGSP{Parameter No }^** **Parameters;** **Parameter No:**Specifies parameter number which will be changed. Value must be between 1 -20.
Example	~SPCGSP{1}^ Return Value(On Successed) : ~ SPGRES{SPCGSP:25}^ -- First system parameter is 25.

2.16 Set/Get All System Parameters

SPCSPA : Allows to set all system parameters. Savema Printers have 20 pieces of parameters and each parameter function is changing according to printer type. All parameters must be sent in proper sequence. Otherwise printer doesn't apply this parameters. Please look at System Paramaters Explanation for more info at the end of this document.

Printer sends OK message when setting all system parameters is successed or sends FAIL message when settingall system parameters is failed.

Using	**~SPCSPA{P1>P2>P3>P4>P5>P6>P7>P8>P9>P10>P11>P12>P13>P14>P15>P16>P17> P18>P19>P20}^** P means Parameter **Parameters;** **Parameter Values :**Specifies parameter values of all system parameters. Minimum and maximum value is changing according to parameter function and printer type. Please look at System Paramaters Explanation for more info at the end of this document. **Note:**Affects printer working, so be carefull while setting parameter values. Return Value(On Successed) : ~ SPGRES{ SPCSPA:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSPA:FAIL}^
Example	~SPCSPA{25>27>300>200>31>77>0>24>25>0>1265>0>5>0>23>0>4>0>0>400}^ - Set all system parameters according to specified values.

SPCGPA : Returnsall system parameters value.

Using	**~SPCGPA^**
Example	~SPCGPA^ Return Value(On Successed) : ~ SPGRES{SPCGPA:25<27<300<200<31<77<0<24<25<0<1265<0<5<0<23<0<4<0<0<400}^ -- All system parameters are returned with SPGRES command.

2.17 Set/Get One Additional Settings

SPCSOA : Allows to set selected additional settings . Additional settings will be used for general purposes.

Printer sends OK message when setting selected additional settings is successed or sends FAIL message when setting selected additional settings is failed.

Using	**~SPCSOA{Parameter No>Parameter value}^** **Parameters;** **Parameter No:**Specifies parameter number which will be changed. Value must be between 1 -20. **Parameter Value :**Specifies parameter value of selected additional settings . This parameter's minimum and maximum value are between 0-3000. Return Value(On Successed) : ~ SPGRES{ SPCSOA:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSOA:FAIL}^
Example	~SPCSOA{1>25}^ - Set First additional setting to 25 ~SPCSOA{15>20}^ - Set 15th additional settings to 20

SPCGOA : Returns selected additional settings value.

Using	**~SPCGOA{Parameter No }^** **Parameters;** **Parameter No:**Specifies parameter number which will be changed. Value must be between 1 -20.
Example	~SPCGOA{1}^ Return Value(On Successed) : ~ SPGRES{SPCGOA:25}^ -- First system parameter is 25.

2.18 Set/Get All Additional Settings

SPCSAA : Allows to set all additional settings. Additional settings have 20 pieces of parameters. All parameters must be sent in proper sequence. Otherwise printer doesn't apply this parameters.

Printer sends OK message when setting all additional settings is successed or sends FAIL message when setting all additional settings are failed.

Using	**~SPCSAA{P1>P2>P3>P4>P5>P6>P7>P8>P9>P10>P11>P12>P13>P14>P15>P16>P17> P18>P19>P20}^** P means Parameter **Parameters;** **Parameter Values :**Specifies parameter values of all additional settings Minimum and maximum value are between 0-3000. Please look at System **Note:**Affects printer working, so be carefull while setting parameter values. Return Value(On Successed) : ~ SPGRES{ SPCSAA:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSAA:FAIL}^
Example	~SPCSAA{10>20>30>40>50>60>150>300>685>1150>1265>24>890>0>23>100>4>54>32>400}^ - Set all additional settings according to specified values.

SPCGAA : Returns all additional settings value.

Using	**~SPCGAA^**
Example	~SPCGAA^ Return Value(On Successed) : ~ SPGRES{SPCGAA:10<20<30<40<50<60<150<300<685<1150<1265<24<890<0<23<100<4<54<32<400}^ -- All additional settings are returned with SPGRES command.

2.19 Set /Get System Language

SPCSSL : Allows to change System Interface Language.

Printer sends OK message when setting system language is successed or sends FAIL message when setting system language is failed. Not

Using	**~SPCSSL{System Languge Code }^** **Parameters;** **System Language Code:**Specifies system interface language. Not used codes, can be used later. Now, if language code bigger than 18, printer

	turns system language to English as a default. Codes are shown in below; **01 :** Turkish **02** : English **03** : Arabic **04** : German **05** : Russian **06** : French **07** : Spanish **08** : Italian **09** :Czech **10** :Dutch **11** : Chinese **12** : Korean **13** : Portuguese **14** : Sinhala **15** : Hebrew **16** :Polish **17** : Greek **18** : Persian **19** : Not used . . . **50** : Not used Return Value(On Successed) : ~ SPGRES{ SPCSSL:OK}^ Return Value(On Failed) : ~ SPGRES{ SPCSSL:FAIL}^
Example	~SPCSSL{01}^ -- Set system language to Turkish ~SPCSSL{02}^ --Set system language to English

SPCGSL : Returns system Interface code. This codes are shown in above in SPCSSL command explanation.

Using	**~SPCGSL^**
Example	~SPCGSL^ Return Value(On Successed) : ~ SPGRES{ SPCGSL:02}^ -- System Language is English

2.20 Set/Get Administrator Password

SPCSAP : Allows to set system administrator password. Administrator password provides to be restriched some settings on printer. This password must be numerical. Otherwise printer doesn't allow to change system password.
Printer sends OK message when setting administrator password is successed or sends FAIL message when setting password is failed.

Using	**~SPCSAP{System Password }^** **Parameters;** **System Password:**Specifies system password. This password must be numerical. Return Value(On Successed) : ~ SPGRES{SPCSAP :OK}^ Return Value(On Failed) : ~ SPGRES{SPCSAP :FAIL}^
Example	~SPCSAP{123456}^ - System Password is 123456

SPCGAP: Returns system administrator password.

Using	**~SPCGAP^**
Example	~SPCGAP^ Return Value(On Successed) : ~ SPGRES{SPCGAP:123456}^ -- Administrator password is 123456

2.21 Return to Factory Settings

SPCSFS : Returns all parameters to factory setting. Stored templates and data files doens't delete when return to factory settings. Please be carefull while using this command because all parameters deletes and load factory settings.
Printer sends OK message when printer is returned to factory settings or sends FAIL message when return to factory settings is failed.

Using	**~SPCSFS^** Return Value(On Successed) : ~ SPGRES{SPCSFS:OK}^ Return Value(On Failed) : ~ SPGRES{SPCSFS:FAIL}^
Example	~SPCSFS^ - System retuns to factory settings

2.22 Set/Get Print Request Message

SPCSPM : This command provides printer to send message per print after print finished. Printer doesn't send message per print as a default but if you activate this function with this command, printer sends message which is identified by you end of each print..

Using	**~SPCSPM{Print Request Active|Passive>Print Message}^** **Parameters;** **Print Request Active|Passive:**Specifies print request is active or passive. İt can be 0 or 1. **0:** Printer doesn't send print message. This is default value. **1**: Printer sends message end of each print to connected device. **Print Message :**Specifies print message which be send to connected device. Print reques Message is OK as a default. It can be READY or another message. The message length should not exceed 10 characters. Return Value(On Successed) : ~ SPGRES{SPCSPM:OK}^ Return Value(On Failed) : ~ SPGRES{SPCSPM:FAIL}^
Example	~SPCSPM{0>OK}^ - Printer doesn't send message. ~SPCSPM{1>OK}^ - Printer sends ~ SPGRES{OK}^ message end of each print. ~SPCSPM{1>READY }^ - Printer sends ~ SPGRES{READY}^ message end of each print.

SPCGPM: Returns print request situation and print message.

Using	**~SPCGPM^**
Example	~SPCGPM^ Return Value(On Successed) : ~ SPGRES{SPCGPM:0<OK}^ --Print Request message is disabled.

3. Label Designing Commands

3.1 CreateTemplate Datas and Template Structure

SPLTDS : This command creates template on printer side. This command parameter contains whole template data, so this command's parameter can be very long. When this command is sended, printer save this template to its memory.
Template data structure is created in xml format and this structure occurs two parts. First part contains general template datas. This datas specifies template general properties(Name, Printer Type, Width, Height).
Second part contains object datas. This datas specifies object type, object name, X,Y position, rotation, font and specific object datas.
Templates data structure is defined in below. Please look at in there.
Printer sends OK message when creating templateoperation is successed or sends FAIL message when setting creating templateoperation is failed.
Note : Template data transfer time changes according to communication type and speed. If template data is big, transfer time increases.

Using	**~SPLTDS{Template Data}^** **Parameters;** **Template Datas:**Creates whole template content. And this is created in XML format. This can be very long. **Note:** Template data must be created according to template data structure rules. Otherwise it doesn't created Return Value(On Successed) : ~ SPGRES{SPLTDS:OK}^ Return Value(On Failed) : ~ SPGRES{SPLTDS:FAIL}^
Example	~SPLTDS{<Template> <General> <MachineType>53x70I</MachineType> <Name>temp1_53.ronx</Name> <Width>640</Width> <Height>480</Height> </General> <Object> <ObjectType>Text</ObjectType> <Name>text1</Name> <X>10</X> <Y>63</Y>

```
<W>105</W>
<H>33</H>
<Rotate>180</Rotate>
<Hidden>False</Hidden>
<Content>
    <Data>savema Printer</Data>
    <Source>Internal</Source>
<MagnificationRatio>100</MagnificationRatio>
    <Inverted>False</Inverted>
</Content>
<Font>
    <Name>Arial</Name>
    <Size>15</Size>
    <Style>Bold,Italic</Style>
</Font>
</Object>
</Template>}^   --- This template has only one text
```

3.1.1 General Template Datas

This settings does not affect objects and only related with template general settings. Our machine Print Head's resolution is **300 Dpi**. So measurements are determining according to **300 Dpi.** Template settings are shown in below ;

1- **MachineType :** Specifies type of machine.Machine types affects width and maximum height of template. Machine types are;
   a- **32x40I :** This is Magnetic Intermittent model(without air) and this machine print size must be *Width = 384px* and *Height =max 480px (40 mm)*
   b- **32x50I :** This is Intermittent model and this machine print size must be *Width = 384px*and *Height =max 600px (50 mm)*
   c- **32x70I :** This is Intermittent model and this machine print size must be *Width = 384px*and *Height =max 840px (70 mm)*
   d- **32C :** This is Continuous model and this machine print size must be *Width = 384px* and *Height = max 1500px (125 mm).*
   e- **32C with Cassette :** This is Continuous model and this machine print size must be *Width = 384px* and *Height = max 1500px (125 mm).*
   f- **32x250C :** This is Continuous model and this machine print size must be *Width = 384px* and *Height = max 3000px (250 mm).*
   g- **32x500C :** This is Continuous model and this machine print size must be *Width = 384px* and *Height = max 6000px (500 mm).*

h- **53x40I :** This is Magnetic Intermittent model(without air) and this machine print size must be *Width = 640px* and *Height = max 480px (40 mm)*

i- **53x50I :** This is Intermittent model and this machine print size must be *Width = 640px*and *Height = max 600px (50 mm)*

j- **53x70I :** This is Intermittent model and this machine print size must be *Width = 640px*and *Height = max 840px (70 mm)*

k- **53x125I :** This is Intermittent model and this machine print size must be *Width = 640px*and *Height = max 1500px (125 mm)*

l- **53C :** This is Intermittent model and this machine print size must be *Width = 640px* and *Height = max 1500px (125 mm)*

m- **53x250C :** This is Continuous model and this machine print size must be *Width = 640px* and *Height = max 3000px (250 mm).*

n- **53x500C :** This is Continuous model and this machine print size must be *Width = 640px* and *Height = max 6000px (500 mm).*

o- **107x75I :** This is Intermittent model and this machine print size must be *Width = 1280px*and *Height = max 900px (75 mm)*

p- **107x125I :** This is Intermittent model and this machine print size must be *Width = 1280px* and *Height = max 1500 (125 mm)*

q- **107C :** This is Intermittent model and this machine print size must be *Width =1280px* and *Height = max 1500px (125 mm)*

r- **107x250C :** This is Intermittent model and this machine print size must be *Width =1280px* and *Height = max 3000px (1280 mm)*

s- **TR32 :** This is Traverse model and print sizemust be *Width = 384px* and *Height = max 1500px (125 mm)*

t- **TR53 :** This is Traverse model and print sizemust be *Width = 640px* and *Height = max 1500px (125 mm)*

u- **TR107 :** This is Traverse model and print sizemust be *Width = 1280px* and *Height = max 1500px (125 mm)*

2- **Name :** Specifies name of template file.This name format is name**_53[32,107].ronx. For example temp1_53.ronx, temp1** is name**, _53** shows machine printhead widthand it can be **32** and **107, .ronx** is an extension of file name.

3- **Width :** Specifies width of template**.** This is pixel value and 1 mm = 12 pixels(300 dpi) . It can be 384(32mm), 640(53mm), 1280(107mm).

4- **Height :** Specifies height of template. This is pixel value and 1 mm = 12 pixels(300 dpi). It changes minimum 12 pixel and maximum 1500 pixels. But maximum height is changing according to machine type.(For machine type, look at MachineType property.)

3.1.2 General Object Datas

Some properties are using for all objects.

1- **ObjectType :** Specifies type of object. These are shown in below;
   a- **Date**
   b- **Time**
   c- **Text**
   d- **Counter**
   e- **Logo**
   f- **Shape**
   g- **Barcode**
   h- **2Dbarcode**
2- **Name :** Specifies name of object. Each object name must be different from the others. This names must be increased sequentally. For example date1,date2,..text1,text2..etc.

3- **X :** Specifies X(horizontal) position of object. This is pixel value and 1 mm=12 pixels (300 dpi).

4- **Y:** Specifies Y(vertical) position of object. This is pixel value and 1 mm= 12 pixels(300 dpi).

5- **W :** Specifies width of object. This is pixel value and 1 mm= 12 pixels(300 dpi).

6- **H :** Specifies height of object. This is pixel value and 1 mm= 12 pixels(300 dpi).

7- **Rotate :** Specifies rotation of object. Default it is 0. There are ;
   a- **0 :** Default rotation of object.
   b- **90 :** It turns object to clockwise.
   c- **180 :** It turns object to reverse.
   d- **270 :** It turns object to anticlockwise.
8- **Hidden** : Specifies visibility of object. Values can be True or False. Default value is False.
   a. True : Object will be shown in printer controller but will be not printed.
   b. False : Object will be shown in printer controller and on print.

3.1.3 Objects(Content)

3.1.3.1 Date

Date object have various properties which is shown in below,

1- **Data** : This item stores date value. For example 21.01.2017

2- **Format**: This item stores format of date. Date object support many of different format. Generally it combines around these format type. For example for 21.01.2017 , format items value is shown in ( ).
   a- **dd** : Short day value. (21)
   b- **ddd** : Tree letter day value. (Sat)
   c- **dddd** : Long day value. (Saturday)
   d- **MM** : Short month value.(01)
   e- **MMM** : Tree letter month value. (Jan)
   f- **MMMM** : Long month value. (January)
   g- **yy** : Short year value. (17)
   h- **yyyy** : Long year value. (2017)
   i- **jjj** : Julian Date, Day of year . (021)
   j- **yjjj**: Last digit of year and Julian date(7021)
   k- **jjjy** : Julian date and last digit of year(0217)
   l- **DoW** :Day sequence in a week. (6)
   m- **WWW** : Week sequence in a year. (4)

   For example if you use dd-MM-yyyy format, date appears 21.01.2017

3- **Separator** : Stores separator which seperates date values.You can see it below;
   For example for 21.01.2017 ,
   a- **Space ( )** : 21 01 2017
   b- **Slash (/)** : 21/01/2017
   c- **Back Slash (\)** : 21\01\2017
   d- **Dot (.)** : 21.01.2017
   e- **Comma (,)** : 21,01,2017
   f- **Hypen (-)** : 21-01-2017
   g- **Colon (:)** : 21:01:2017
   h- **None ()** : 21012017

4- **CountryCode** : This item stores language country code for date. It is using date presentation.
   This code is changing according to country. Default language country is English(USA) and CountryCode is 1033 . You can find more info from this link: https://msdn.microsoft.com/en-us/library/ee825488%28v=cs.20%29.aspx.
   Country codes are shown under culture code. Culture code is shown as hexadecimal format. So, it must be convert to decimal format. Englsih- United States culture code is 0x0409 is table. Hex 0x0409 is same is 1033 in decimal format.

5- **DayOffset** : Specifies how many days will be add on actual date.

6- **MonthOffset** : Specifies how many months will be add on actual date.

7- **YearOffset** : Specifies how many years will be add on actual date.

8- **Type** : There are two type of dates.
   a- **Actual** : it changes automatically according to system date.
   b- **Fixed** : it doesn't change according to system date. İt stores only saved date data from PC.

9- **UpperCase** : Specifies date character as Upper or lower
   a- **True** : Date shown as 21 JAN 2017
   b- **False** : Date shown as 21 Jan 2017

10- **UseSpecialMonthNames :** Provides to use special names instead of standart Month names.

11- **SpecialMonthNames :** Contains special month names. Month names are 12 different words or letters and moth names must be seperated with – character.

Example

```
~SPLTDS{<Template>
        <General>
                <MachineType>53x70I</MachineType>
                <Name>temp1_53.ronx</Name>
                <Width>640</Width>
                <Height>480</Height>
        </General>
        <Object>
                <ObjectType>Date</ObjectType>
                <Name>date1</Name>
                <X>20</X>
                <Y>4</Y>
                <W>146</W>
                <H>33</H>
                <Rotate>0</Rotate>
                <Hidden>False</Hidden>
                <Content>
                        <Data>21/01/2017</Data>
                        <Format>dd/MM/yyyy</Format>
                        <Separator>/</Separator>
                        <CountryCode>1033</CountryCode>
                        <DayOffset>0</DayOffset>
                        <MonthOffset>0</MonthOffset>
                        <YearOffset>0</YearOffset>
```

```
                    <Type>Actual</Type>
                    <UpperCase>True</UpperCase>
      <UseSpecialMonthNames>True</UseSpecialMonthNames>
  <SpecialMonthNames>A-B-C-D-E-F-G-H-I-J-K-L
</SpecialMonthNames>
            </Content>
             <Font>
                    <Name>Arial</Name>
                    <Size>20</Size>
                    <Style>Bold</Style>
             </Font>
        </Object>
    </Template>}^   --- This template has only one date
```

3.1.3.2 Time

Time object have various properties which is shown in below,

1- **Data** : This item stores time value.For example 15:23
2- **Format**: This item stores format of time. Time object support many of different format. Generally it combines around these format type. For example for 15:23:00 , format items value is shown in ( ).
   a- **HH** : Hour value according to 24 hours . (15)
   b- **hh** : Hour value according to 24 hours . (03)
   c- **mm** : Minute value (23)
   d- **ss** : Second value (00)
   e- **tt** : Time symbol(AM/PM).It changes according to some country. (03:23 PM)

   For example if you use hh:mm tt format, time appears 03:23 PM

3- **Seperator** : This item stores seperator which seperates time values.You can see it below;For example for 15:23,
   a- **Space ( )** : 15 23
   b- **Slash (/)** : 15/23
   c- **Back Slash (\)** : 15\23
   d- **Dot (.)** : 15.23
   e- **Comma (,)** : 15,23
   f- **Hypen (-)** : 15-23
   g- **Colon (:)** : 15:23 - Default
   h- **None ()** : 1523
4- **CountryCode** : This item stores language country code for time. It is using time presentation. Especially it is useful for time symbol. if English(USA) is selected, time

symbol is AM/PM. This code is changing according to country. Default language country is English(USA) and CountryCode is 1033 . You can find more info from this link:
https://msdn.microsoft.com/en-us/library/ee825488%28v=cs.20%29.aspx.
Country codes are shown under culture code. Culture code is shown as hexadecimal format. So, it must be convert to decimal format. Englsih- United States culture code is 0x0409 is table. Hex 0x0409 is same is 1033 in decimal format.

5- **HourOffset** : Specifies how many hours will be add on actual time.
6- **MinuteOffset** : Specifies how many minutes will be add on actual time.
7- **Type** : There are two type of dates.
   a- **Actual** : it changes automatically according to system time.
   b- **Fixed** : it doesn't change according to system time. İt stores only saved time data from PC.

Example

```
~SPLTDS{<Template>
        <General>
                <MachineType>53x70I</MachineType>
                <Name>temp1_53.ronx</Name>
                <Width>640</Width>
                <Height>480</Height>
        </General>
        <Object>
                <ObjectType>Time</ObjectType
                <Name>time1</Name>
                <X>203</X>
                <Y>0</Y
                <W>77</W>
                <H>33</H>
                <Rotate>0</Rotate>
                <Hidden>False</Hidden>
                <Content>
                        <Data>15:23</Data>
                        <Format>HH:mm</Format>
                        <Separator>:</Separator>
                        <CountryCode>1033</CountryCode>
                        <HourOffset>0</HourOffset>
                        <MinuteOffset>0</MinuteOffset>
                        <Type>Actual</Type>
                </Content>
                <Font>
                        <Name>Arial</Name>
```

	<Size>20</Size> <Style>Regular</Style> </Font> </Object> </Template>}^ --- This template has only one text

3.1.3.3 Text

Text object have two different properties which is shown in below,

1- **Data** : This item stores text value which is saved by PC. It uses ~ character for seperating lines for multiline text. For example if this item value is savema~printer, printer shows
savema
printer
in screen. According to this sample, *savema* is first line, printer is second line.

2- **Source** : This item specify text value source. it can be Internal and External.
   a- **Internal** : This is default selection for text. Text value is identified from PC when creating a template in this mode.
   b- **External** : Text object gets value from RS-232 or Ethernet interface.

3- **MagnificationRatio** : This item stores magnification ratio of width. This ratio is shown in percentage. For Normal width This value must be 100 (100%). For bigger width of text, this value must be bigger than 100 and it can change according to magnification.

4- **Inverted** : Text will be print inverted when adjus this item True. This item is set False as a default.

Example	~SPLTDS{<Template> <General> <MachineType>53x70I</MachineType> <Name>temp1_53.ronx</Name> <Width>640</Width> <Height>480</Height> </General> <Object> <ObjectType>Text</ObjectType> <Name>text1</Name> <X>10</X>

	<Y>63</Y> <W>105</W> <H>33</H> <Rotate>180</Rotate> <Hidden>False</Hidden> <Content> <Data>savema Printer</Data> <Source>Internal</Source> <MagnificationRatio>100</MagnificationRatio> <Inverted>False</Inverted> </Content> <Font> <Name>Arial</Name> <Size>15</Size> <Style>Bold,Italic</Style> </Font> </Object> </Template>}^ --- This template has only one text

3.1.3.4 RichText

RichText object have two different properties which is shown in below,

1- **RtfData** : This item stores rtf value of entered richtext which is saved by PC Software.
2- **ImageData**: This item storages RichText image's string in Base64 standart. Images are must be convert to Base64 data for this parameter.

Example	~SPLTDS{<Template> <General> <MachineType>53x70I</MachineType> <Name>temp1_53.ronx</Name> <Width>640</Width> <Height>480</Height> </General> <Object> <ObjectType>RichText</ObjectType> <Name>richtext1</Name> <X>10</X>

<Y>63</Y>
<W>105</W>
<H>33</H>
<Rotate>180</Rotate>
<Hidden>False</Hidden>
<Content>
<RtfData>{\rtf1\ansi\ansicpg1254\deff0\deflang1055{\fonttbl{\f0\fnil\fcharset162{*\fname Arial;}Arial TUR;}}
\viewkind4\uc1\pard\f0\fs41 Savema \b\i thermal \b0 transfer\i0\par
}</RtfData>
<ImageData>iVBORw0KGgoAAAANSUhEUgAAATwAAAAUCAYAAAAQjYHpAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAe+SURBVHhe7Vrbd
****
Multiple Lines deleted
****
2cKJITXfWRFZsVs3lHI5Zvaes8Zz+POziw0PsuzXrHkNDgwvZFZ+8pSOzFehKe8JcApxLvWu3kxTyfuHsYlXf/IThQ6uhx2OBpreUofUXaX5GKF/RWjcddY0n0NNZmM7sMI3wty39ZJV6IYXnusQAAAABJRU5ErkJggg==</ImageData>
</Content>
<Font>
<Name>Arial</Name>
<Size>15</Size>
<Style>Bold,Italic</Style>
</Font>
</Object>
</Template>}^ --- This template has only one richtext

3.1.3.5 Counter

Counter object have various properties which is shown in below,

1- **CounterType** : This item specify type of counter. There are three counter types which is used.
   a- **Numeric** : it uses only numbers. For example, 1,2,3…999..etc
   b- **Alphabetic** : it uses only alpha. For example A,B,C,…ZZ…etc.

c- **AlphaNumeric** : it combines alpha and numbers with together. Numbers are changing firstly and after end of numbers alpha is changing .For example AA000,AA001,…,AA999,AB000,AB001…..ZZ999

2- **IncreasingDecreasing** : it shows counter value is increasing or decreasing. It can be Increasing or Decreasing.
   a- Increasing : Counter starts small value and goes to big value. For example, 001,002…999
   b- Decreasing : Counter starts big value and goes to smal value. 999,998,…002,001

3- **Data:** Counter value

4- **NumericBegin :** Numeric counter beginning value. For example 0000.

5- **NumericEnd :** Numeric counter ending value. For example 9999.
6- **NumericStep :** Numeric counter step value. Default it is 1. For 1, counter increases or decrease one by one. For example,1,2,3…999

7- **NumericPeriod :** Numeric counter period value. it shows counter increase or decrease after how many print. For example if this value is 3, printer prints same value 3 times. (1,1,1,2,2,2)

8- **NumericDigit :** Counter numeric digit. For example if it is 4, counter value show 4 digits.

9- **AlphaBegin :** Alphabetic counter beginning value. For example AAA

10- **AlphaEnd :** Alphabetic counter ending value. For example ZZZ

11- **AlphaStep :** Alphabetic counter step value. Default it is 1. For 1, counter increases or decrease one by one. For example,AAA,AAB…..ZZZ

12- **AlphaPeriod :** Alphabetic counter period value. it shows counter increase or decrease after how many print. For example if this value is 3, printer prints same value 3 times. (AAA,AAA,AAA,AAB,AAB,AAB……ZZZ,ZZZ,ZZZ)
13- **AlphaDigit :** Counter alpha digit. For example if it is 3, counter value show 3 digits. (AAA)
14- **AlphaChar :** Padding character for alphabetic counter.it is using before counter value. Default it is A.So, counter shows AAA instead of A.
15- **Restart** : It can be True or False. When counter value arrives end value, it shows this value is turn to beginning value or not. (stay last value).

Example

```
~SPLTDS{<Template>
        <General>
                <MachineType>53x70I</MachineType>
                <Name>temp1_53.ronx</Name>
                <Width>640</Width>
                <Height>480</Height>
        </General>
        <Object>
                <ObjectType>Counter</ObjectType>
                <Name>counter1</Name>
                <X>10</X>
                <Y>63</Y>
                <W>100</W>
                <H>33</H>
                <Rotate>0</Rotate>
                <Hidden>False</Hidden>
                <Content>
                        <CounterType>Numeric</CounterType>
        <IncreasingDecreasing>Increasing</IncreasingDecreasing>
                        <NumericBegin>000000</NumericBegin>
                        <NumericEnd>999999</NumericEnd>
                        <NumericStep>1</NumericStep>
                        <NumericPeriod>1</NumericPeriod>
                        <NumericDigit>6</NumericDigit>
                        <AlphaBegin>AAAAAA</AlphaBegin>
                        <AlphaEnd>ZZZZZZ</AlphaEnd>
                        <AlphaStep>1</AlphaStep>
                        <AlphaPeriod>1</AlphaPeriod>
                        <AlphaDigit>6</AlphaDigit>
                        <AlphaChar>A</AlphaChar>
                        <Restart>True</Restart>
                </Content>
                <Font>
                        <Name>Arial</Name>
                        <Size>20</Size>
                        <Style>Bold</Style>
                </Font>
        </Object>
    </Template>}^   --- This template has only one counter
```

3.1.3.6 Logo(Image)

Logo object have two different properties which is shown in below,

1- **ImageData** : This item storages image's string in Base64 standart. Images are must be convert to Base64 data fort his parameter.

Example


```
~SPLTDS{<Template>
	<General>
		<MachineType>53x70I</MachineType>
		<Name>temp1_53.ronx</Name>
		<Width>640</Width>
		<Height>480</Height>
	</General>
	<Object>
		<ObjectType>Logo</ObjectType>
		<Name>logo1</Name>
		<X>20</X>
		<Y>20</Y>
		<W>30</W>
		<H>42</H>
		<Rotate>0</Rotate>
		<Hidden>False</Hidden>
		<Content>
<ImageData>/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgcGB
QgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0aHBwgJC4nICIsIxw
cKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyI
RwhMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMj
IyMjIyMjIyMjL/wAARCAAqAB4DASIAAhEBAxEB/8QAHwAAAQUB
AQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIE
AwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwR
VS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFV
WV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZm
qKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+
Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAA
AAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECA
xEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOE
l8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2
hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmq
srO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8v
P09fb3+Pn6/9oADAMBAAIRAxEAPwD3+vE/2evEmta5b+IINW1S7
v0tnt3hN1KZGQuJA2GbJwdi8ZwOcdTnvPEXxR8HeGPMjvtagluk8
```

	wG1tD50m9OqELkI2ePnK8554OPnz4Tav4r0+31618LWdoHuEie7 1O9cLBYRKJAZGJwMjfu5zxG3ytzgA+s6K+WPEN1Z6heLaeKvG2q+ LNSm/wBXpnh4D7MtxgCLbIw2HcrYPlxEgls8j5vqegD5k8L6Xqep Wqf8K78Dx28f2hSviLXDHPLgSnZJHvUIhXDBhGrngfxAE5/wX+Hmi +O7jVZdZe7Kac9uyRQyBFlDF9yvwTg7APlIPJ59PquvAP2Zf+Zp/wC 3T/2tQB7HoHg7w54WRV0XR7S0cIU85U3SspbcQ0jZdhnHBJ6D0F cn4p+Nfhvwj4ju9Dv7LVZLq12b3gijKHciuMEyA9GHau41zWbPw9 od7q9++y1tImlfBALY6KuSAWJwAM8kgV4R8FvDSeM/EPiLxhr2n WNza3Eska281ussZnkcSuVDEldowBkHIk68GgD6Hr5E+GPxLh+HV vq7NpcmoT3zwBEEwiVVQSbiWwxzl1wMevIxz9d1j2PhPw3pl5He WHh/SrS6jzsmgso43XIIOGAyMgkfjQB4ZZaJ4/8AjTf2dx4m8zTPC6 OJ1WOPylb5F5iRss5YHh23KNz4P8J9/wBK0qx0TS7fTNMto7azt02 RRIOFH8ySckk8kkk8mrlFAH//2Q== </ ImageData> </Content> <Font> <Name>Arial</Name> <Size>15</Size> <Style>Bold</Style> </Font> </Object> </Template>}^ --- This template has only one logo

3.1.3.7 Shape

Shape object have two different properties which is shown in below,

1- **ShapeType**: This item specifies shape type. This can be Rectangle, Ellipse, FilledRectangle and FilledEllipse.
2- **LineThickness :** This item adjust thickness of line. It is measured in pixel.Forexample; 12 pixels = 1mm**.**Line thickness cannot be bigger than widht and height value.

Example	~SPLTDS{<Template> <General> <MachineType>53x70I</MachineType> <Name>temp1_53.ronx</Name> <Width>640</Width> <Height>480</Height> </General> <Object> <ObjectType>Shape</ObjectType>

	<Name>shape1</Name> <X>20</X> <Y>20</Y> <W>120</W> <H>120</H> <Rotate>0</Rotate> <Hidden>False</Hidden> <Content> <ShapeType>Rectangle</ShapeType> <LineThickness>12</LineThickness> </Content> <Font> <Name>Arial</Name> <Size>15</Size> <Style>Bold </Style> </Font> </Object> </Template>}^ --- This template has only one shape

3.1.3.8 Shift Code

Shift code object have various properties which is shown in below,

1- **ShiftNo**: Shows shift count in a day. Minimum value is 1, maximum value is 6.

2- **Shift1_Start**: First Shift start time.

3- **Shift1_Text:** First Shift text. Text is shown between Shift1 start and stop time.

4- **Shift2_Start**: Second Shift start time. If you don't use second shift, it must be 00:00.

5- **Shift2_Text:** Second Shift text. Text is shown between Shift2 start and stop time.if you don't use second shift, it can be blank.

6- **Shift3_Start**: Third Shift start time. If you don't use third shift, it must be 00:00.

7- **Shift3_Text:** Third Shift text. Text is shown between Shift3 start and stop time. If you don't use second shift, it can be blank.

8- **Shift4_Start**: Fourth Shift start time. If you don't use fourth shift, it must be 00:00.

9- **Shift4_Text:** Fourth Shift text. Text is shown between Shift4 start and stop time. If you don't use fourth shift, it can be blank.

10- **Shift5_Start**: Fifth Shift start time.If you don't use fifth shift, it must be 00:00.

11- **Shift5_Text:** Fifth Shift text. Text is shown between Shift5 start and stop time. If you don't use second shift, it can be blank.

12- **Shift6_Start**: Sixth Shift start time. If you don't use fifth shift, it must be 00:00.

13- **Shift6_Text:** Sixth Shift text. Text is shown between Shift6 start and stop time. If you don't use sixth shift, it can be blank..

14- **MaxCharNo**: Specifies maximum usable count of characters which will use in Shift Texts.

Example

```
~SPLTDS{<Template>
        <General>
                <MachineType>53x70I</MachineType>
                <Name>temp1_53.ronx</Name>
                <Width>640</Width>
                <Height>480</Height>
        </General>
        <Object>
                <ObjectType>Shift</ObjectType>
                <Name>shift1</Name>
                <X>38</X>
                <Y>348</Y>
                <W>95</W>
                <H>33</H>
                <Rotate>0</Rotate>
                <Hidden>False</Hidden>
                <Content><!-- Shift object properties-->
                        <ShiftNo>3</ShiftNo>
                        <Shift1_Start>00:00</Shift1_Start>
                        <Shift1_Text>SHFT1</Shift1_Text>
                        <Shift2_Start>08:00</Shift2_Start>
                        <Shift2_Text>SHFT2</Shift2_Text>
                        <Shift3_Start>16:00</Shift3_Start>
                        <Shift3_Text>SHFT3</Shift3_Text>
```

	<Shift4_Start>00:00</Shift4_Start> <Shift4_Text></Shift4_Text> <Shift5_Start>00:00</Shift5_Start> <Shift5_Text></Shift5_Text> <Shift6_Start>00:00</Shift6_Start> <Shift6_Text></Shift6_Text> <MaxCharNo>5</MaxCharNo> </Content> <Font> <Name>Arial</Name> <Size>20</Size> <Style>Regular</Style </Font> </Object> </Template>}^ --- This template has only one shift code

3.1.3.9 Barcode

Barcode object have various properties which is shown in below,

1- **Source**: This item specify barcode value source. it can be Internal and External.
   a- **Internal** : This is default selection for barcode. Barcode value is identified from PC when creating a template in this mode.
   b- **External** : Barcode object gets value from RS-232 or Ethernet interface.
   c- **Counter** : Barcode object value is increasing one by one. Barcode values must be numeric in this type.
   d- **Database** : Barcode object value is changing one by one with data file(.csv file) content. Csv file must be stored in controller.

2- **BarcodeType :** It stores barcode type. We support many barcodes. These are;
   - **Codabar**
   - **Code 11**
   - **Code 128**
   - **Code 32**
   - **Code 39**
   - **Code 93**
   - **Deutsche Post Identcode**
   - **Deutsche Post Leitcode**
   - **EAN-13**

- **EAN-8**
- **EAN-99**
- **EAN-Velocity**
- **FedEx Ground 96**
- **Industrial 2 of 5**
- **Interleaved 2 o5**
- **ISBN**
- **ISMN**
- **ISSN**
- **ITF-14**
- **JAN-13**
- **JAN-8**
- **MSI**
- **OPC**
- **PharmaCode**
- **PLANET**
- **POSTNET**
- **PZN**
- **SCC-14**
- **SCC-18**
- **Telepen**
- **UCC/EAN-128**
- **GS1-128**
- **UPC-A**
- **UPC-E**
- **GS1-Databar Omnidirectional**
- **GS1-Databar Omnidirectional Stacked**
- **GS1-Databar Truncated**
- **GS1-Databar Limited**
- **GS1-Databar Stacked**
- **GS1-Databar Expanded**
- **GS1-Databar Expanded Stacked**

3- **BarcodeValue:** It stores barcode value.Barcode value must be suitable for barcode type. For example 8691234567890 value is suitable for EAN-13 but 8691234567891 is not suitable.

4- **AddCheckSum :** Specifieswhether checksum must be generated and attached to the value to encode.This property is using for some barcodes.(Code 128, Code 39...etc) . It can be True or False.

a- **True** : Adding checksum value to barcode

b- **False** : Do not add cheksum value to barcode

5- **BarHeight :** Specifiesthe bar's height of the barcode. It values are changing between 0.1-5.0 values. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

6- **BarRatio :** Specifiesthe wide bar's width compared to the narrow bar's width. In barcode terminology this is N value. It is changing between 1-20.

7- **BarWidth :** Specifies the narrow bar's width of the barcode. In barcode terminology this is X value. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

8- **BearerBarStyle :** Specifiesthe bearer bar's type that must be drawn with the barcode image. Bearer bar is only available for 2 of 5, Code 128 and UCC/EAN-128 barcodes**.** It can beNone**,** FrameandHorizontal Rules**.** These are;

a- **None :** it doesn't add any line around the barcode
b- **Frame :** it adds frame to around the barcode
c- **Horizontal Rules :** it adds lines top and bottom of barcode lines.

9- **BearerBarWidth :**Specifies the bearer bar's width. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

10- **BorderWidth :**Specifiesthe barcode image border's width. Border property is drawing frame around barcode. Default it is 0. If it is 0, cannot see any frame around barcode.

11- **CodabarStartChar :**Specifiesthe start character for Codabar symbology. Possible values are: A, B, C or D.

12- **CodabarStopChar :** Specifiesthe stop character for Codabar symbology. Possible values are: A, B, C or D.

13- **Code128Charset :** Specifies the characters set to use in the Code 128 symbology. Possible values are: Auto, A, B or C.

14- **CodeAlignment :** Specifies location of code according to barcode. It can be Below Left, Below Center, Below Right, Above Left, Above Center, Above Right.

15- **DisplayCode :** Specifieswhether the value to encode must be displayed in the barcode image. It can be True or False.

a- **True :** Barcode value appears with barcode value.

b- **False :** Barcode value code doesn't appear.

16- **DisplayChecksum :** Specifies whether checksum is printed or not.It can be True or False.

c- **True :** Checksum appears with barcode value.

d- **False :** Checksum value code doesn't appear. This value is default.

17- **DisplayLightMarginIndicator :** Specifieswhether light margin indicators must be displayed in the barcode image. Only available for EAN/UPC Symbologies. It can be True or False.

a- True : Indicator appears with barcode..

b- False : Indicator code doesn't appear.

18- **DisplayStartStopChar :** Specifieswhether start and stop characters must be displayed in the barcode image. İt is usign with some barcodes(Codabar, Code39..etc). It can be True or False.

a- True : Start Stop Charappears with barcode value.

b- False : Start Stop Char doesn't appear.

19- **EanUpcSupplementType :** Specifies whether use EAN or UPC with supplement barcode or use single type barcode. It is using with EAN and UPC barcodes(EAN13, EAN8, UPC-A, UPC-E). It can be None, Digits2 (Addon 2) or Digits5 (Addon 5). Addon barcode values doesn't read by barcode scanner.

a- None : Only single barcode appears

b- Digits2 : Barcode appears with two digits addon barcode

c- Digits5: Barcode appears with five digits addon barcode

20- **EanUpcSupplementCode :** Specifies value of supplement barcode. If supplement type is Digits2, this value must be 2 digits. If supplement type is Digits5, value must be 5 digits. Code value must be numerical.

21- **EanUpcSupplementSeparator :** Specifies distance between main barcode bars and addon(supplement) barcode bars. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

22- **EanUpcSupplementMargin :** Specifies distance between top of addon barcode bars and addon value. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

23- **SegmentCount :** Specifies segment count of GS1-Databar Expanded and GS1-Databar Expanded Stacked. This is numerical value and default it is 6. It can be 2,4,6,8,10,12,14,16,18 and 20.

24- **QuietZoneWidth :** Specifies the right and left side gap of the barcode. It values are changing between 0.01 - 5.0 values. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

25- **PharmaCodeBarSpacing:** Specifies gaps between bars of PharmaCode barcode. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

26- **PharmaCodeThickBarWidth:** Specifies the thick bar width of the PharmaCode barcode. It values are changing between 0.01 - 5.0 values. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

27- **PharmaCodeThinBarWidth:** Specifies the thin bar width of the PharmaCode barcode. It values are changing between 0.01 - 5.0 values. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

28- **ShortBarHeight:** Specifies the short bar height of the PLANET and POSTNET barcode. It values are changing between 0.01 - 5.0 values. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

29- **TallBarHeight:** Specifies the tall bar height of the PLANET and POSTNET barcode. It values are changing between 0.01 - 5.0 values. It's measured in inches. Please see note at the end of barcode object. It shows inch calculating.

30- **TelepenEncoding:** Specifies encoding type of telepen barcode. It can be Ascii or Numeric.

   a. **Ascii :** Allows to use ascii(numeric+alphabetic) characters.
   b. **Numeric :** Allows to use only numeric characters.

31- **UpceSystem :** Specifiesthe number system to use for UPC-E symbology. It can be System0 and System1.

   a- **System0**
   b- **System1**

32- **Text :** Specifies text for barcode. Text doesn't encode into barcode. İt is using only explanation.

33- **Inverted** : If you can print with white ribbon you must select White Ribbon option. The barcode's color inverted automatic. This property is useful when the user prints dark colored pack

34- **CounterBegin :** Specifies counter start value for barcode. This item is only used when Source item adjust as Counter.

35- **CounterEnd:** Specifies counter end value for barcode. This item is only used when Source item adjust as Counter.

36- **CounterStep:** Specifies counters step value. Deafult value is 1. For 1, counter increases one by one. This item is only used when Source item adjust as Counter.

37- **CounterPeriod:** Specifies counter value increase after how many print. Deafult it is 1. This item is only used when Source item adjust as Counter.

38- **CounterDigit:** Specifies digit of counter for counter variable barcode. This item is only used when Source item is Counter.

39- **FileName** : This item stores file name(if possible with path) which keeps data. File must be csv file. İf file name will adjust in printer controller,İt can be default.csv. This item is only used when Source item adjust as database.

40- **ColumnNo** : Specifies column of datas in csv file. İt is default 0. This item is only used when Source item adjust as database.

Note : Barcode component inch standart is 96 dpi but we are using 300 dpi(Print Head's resolution). So, you must be use this formula for calculating original size for savema printer :

Orginal Size = Value*(300/96 dpi) and 1 inch=25,4 mm

For example : For 0.5 value equals 0.5 / (300/96 dpi)= 0.16 inch=4 mm

Example	~SPLTDS{<Template> <General> <MachineType>53x70I</MachineType> <Name>temp1_53.ronx</Name> <Width>640</Width> <Height>480</Height> </General> <Object> <ObjectType>Barcode</ObjectType> <Name>barcode1</Name> <X>11</X> <Y>193</Y>

```
                <W>228</W>
                <H>93</H>
                <Rotate>0</Rotate>
                <Hidden>False</Hidden>
                <Content>
                <Source>Internal</Source>
                <BarcodeType>EAN-13</BarcodeType>
                <BarcodeValue>8691234567890</BarcodeValue>
                <AddCheckSum>True</AddCheckSum>
                <BarHeight>0.7</BarHeight>
                <BarRatio>2</BarRatio>
                <BarWidth>0.02</BarWidth>
                <BearerBarStyle>None</BearerBarStyle>
                <BearerBarWidth>0.02</BearerBarWidth>
                <BorderWidth>0</BorderWidth>
                <CodabarStartChar>A</CodabarStartChar>
                <CodabarStopChar>A</CodabarStopChar>
                <Code128Charset>Auto</Code128Charset>
                <CodeAlignment>BelowCenter</CodeAlignment>
<DisplayCode>True</DisplayCode><DisplayLightMarginIndicator
>False</DisplayLightMarginIndicator>
            <DisplayStartStopChar>False</DisplayStartStopChar>
    <EanUpcSupplementType>None</EanUpcSupplementType>
      <EanUpcSupplementCode>0</ EanUpcSupplementCode>
       <EanUpcSupplementSeparator>0.2</EanUpcSupplementS
eparator>
<EanUpcSupplementMargin>0.4</ EanUpcSupplementMargin>
                <TelepenEncoding>Numeric</TelepenEncoding>
                <UpceSystem>System0</UpceSystem>
                <Text></Text>
                <Inverted>False</Inverted>
                <CounterBegin>000000</CounterBegin>
                <CounterEnd>999999</CounterEnd>
                <CounterStep>1</CounterStep>
                <CounterPeriod>1</CounterPeriod>
                <CounterDigit>6</CounterDigit>
                <FileName>c:\savema.csv</FileName>
                <ColumnNo>0</ColumnNo>
            </Content>
            <Font>
```

	<Name>Arial</Name> <Size>15</Size> <Style>Bold</Style> </Font> </Object> </Template>}^ --- This template has only one barcode

3.1.3.10 2D Barcode

2D Barcode object have various properties which is shown in below,

1- **Source**: This item specify barcode value source. it can be Internal and External.
   a- **Internal** : This is default selection for barcode. 2DBarcode value is identified from PC when creating a template in this mode.
   b- **External** : Multitext object gets value from RS-232 or Ethernet interface.
   c- **Counter** :2D Barcode object value is increasing one by one. 2D Barcode values must be numeric in this type.
   d- **Database** : 2D Barcode object value is changing one by on according to data file(.csv file) content. Csv file must be stored in controller.

2- **TwoDBarcodeType :** It stores barcode type.
3- We support many barcodes. These are;
   1. **Code16k**
   2. **DataMatrix**
   3. **GS1-Datamatrix**
   4. **QRCode**
   5. **Semacode**
   6. **AztecCode**
   7. **Pdf417**
   8. **CompactPdf417**
   9. **MacroPdf417**
   10. **MicroPDF417**

4- **TwoDBarcodeValue:** It stores barcode value.2D Barcode value must be suitable for barcode type.

5- **ErrorCorrection**
   - **AztecCodeErrorCorrection:** Specifies Error Correction Percentage to apply for Aztec Code symbology. Default is 23.

- **Pdf417ErrorCorrectionLevel :** Specifies the Error Correction Level to apply for PDF417 symbology.
  a) **Level0**
  b) **Level1**
  c) **Level2 :** This is default selection.
  d) **Level3**
  e) **Level4**
  f) **Level5**
  g) **Level6**
  h) **Level7**
  i) **Level 8**

- **QRCodeErrorCorrectionLevel:** Specifies the Error Correction Level to apply for QR Code symbology. There are 4 type of Error Correction.
  a) **L :** Approx. 7% of codewords can be restored. Error correction level L is appropriate for high symbol quality and/or the need for the smallest possible symbol.
  b) **M :** Approx. 15% of codewords can be restored. Level M is described as Standard level and offers a good compromise between small size and increased reliability. We are using this encoding type as a default.
  c) **Q :** Approx. 25% of codewords can be restored. Level Q is a High reliability level and suitable for more critical or poor print quality applications.
  d) **H :** Approx. 30% of codewords can be restored. Level H offers the maximum achievable reliability.

6- CodeFormat :

- **AztecCodeFormat:**Specifies the Aztec Code Format to use on that symbology. This property have alot of format. Auto is using as a default. Formats are ;
  - **Auto**
  - **C15X15Compact**
  - **C19X19**
  - **C19X19Compact**
  - **C23X23**
  - **C23X23Compact**
  - **C27X27**
  - **C27X27Compact**
  - **C31X31**
  - **C37X37**

- **C41X41**
- **C45X45**
- **C49X49**
- **C53X53**
- **C57X57**
- **C61X61**
- **C67X67**
- **C71X71**
- **C75X75**
- **C79X79**
- **C83X83**
- **C87X87**
- **C91X91**
- **C95X95**
- **C101X101**
- **C105X105**
- **C109X109**
- **C113X113**
- **C117X117**
- **C121X121**
- **C125X125**
- **C131X131**
- **C135X135**
- **C139X139**
- **C143X143**
- **C147X147**
- **C151X151**

- **DataMatrixFormat :** Specifies the DataMatrix Format to use on that symbology.
  - **Auto :** This is default selection.
  - **C10X10**
  - **C12X12**
  - **C14X14**
  - **C16CX16**
  - **C18X18**
  - **C20X20**
  - **C22X22**
  - **C24X24**
  - **C26X26**
  - **C32X32**

- **C36X36**
- **C40X40**
- **C44X44**
- **C48X48**
- **C52X52**
- **C64X64**
- **C72X72**
- **C80X80**
- **C88X88**
- **C96X96**
- **C104X104**
- **C120X120**
- **C132X132**
- **C144X144**
- **C8X18**
- **C8X32**
- **C12X26**
- **C12X36**
- **C16X36**
- **C16X48**

7- ModuleSize :

- **AztecCodeModuleSize:** Specifies the module sizeof Aztec Code. It's measured in inches. It is changing between 0.01 and 0.3. Please see note at the end of 2D barcode object. It shows inch calculating.
- **DataMatrixModuleSize :** Specifies DataMatrix module size. It's measured in inches.It is changing between 0.01 and 0.3. Please see note at the end of 2D barcode object. It shows inch calculating.
- **QRCodeModuleSize :** Specifiesthe module size. It's measured in inches. Please see note at the end of 2D barcode object. It shows inch calculating.
- **GS1-DatamatrixModuleSize :** Specifies GS1-DataMatrix module size. It's measured in inches.It is changing between 0.01 and 0.3. Please see note at the end of 2D barcode object. It shows inch calculating.

8- Version :

- **MicroPDF417Version:** Specifies the MicroPDF417 version (a predefined combinations of numbers of columns and rows) to be generated.
  - **Auto :** This is default selection.
  - **V1X11**
  - **V1X17**
  - **V1X20**
  - **V1X24**

- **V1X28**
- **V2X8**
- **V2X11**
- **V2X14**
- **V2X17**
- **V2X20**
- **V2X23**
- **V2X26**
- **V3X6**
- **V3X8**
- **V3X10**
- **V3X12**
- **V3X15**
- **V3X20**
- **V3X26**
- **V3X32**
- **V3X38**
- **V3X44**
- **V4X4**
- **V4X6**
- **V4X8**
- **V4X10**
- **V4X12**
- **V4X15**
- **V4X20**
- **V4X26**
- **V4X32**
- **V4X38**
- **V4X44**

- **QRCodeVersion :** Specifiesthe QR Code Version to use on that symbology. Version 1 (21 x 21 modules) to Version 40 (177 x 177 modules) increasing in steps of four modules per side.
  - **Auto :**
  - **V01**
  - **V02**
  - **V03**
  - **V04**
  - **V05**
  - **V06**
  - **V07**
  - **V08**

- **V09**
- **V10**
- **V11**
- **V12**
- **V13**
- **V14**
- **V15**
- **V16**
- **V17**
- **V18**
- **V19**
- **V20**
- **V21**
- **V22**
- **V23**
- **V24**
- **V25**
- **V26**
- **V27**
- **V28**
- **V29**
- **V30**
- **V31**
- **V32**
- **V33**
- **V34**
- **V35**
- **V36**
- **V37**
- **V38**
- **V39**
- **V40**

9- Encoding :

- **DataMatrixEncoding :** Specifies the DataMatrix Encoding to use on that symbology.
  - **a- Auto :** This is default selection.
  - **b- Ascii :** Used to encode data that mainly contains ASCII characters (0-127).
  - **c- C40 :** Used to encode data that mainly contains numeric and upper case characters.
  - **d- Text :** Used to encode data that mainly contains numeric and lower case characters.

e- **Base256 :** Used to encode 8 bit values

- **QRCodeEncoding :** Specifies the QR Code Encoding to use on that symbology.
  - a- **Auto :** This is default selection.
  - b- **Numeric :** Used to encode data that mainly contains numeric characters.
  - c- **AlphaNumeric :** Used to encode data that mainly contains alphanumeric characters.
  - d- **Kanji :** Used to encode data that mainly contains Kanji characters.
  - e- **Byte :** Used to encode 8 bit values.

10- **AztecCodeRune:** Specifies the Aztec Code Rune value. It must be a value from 0 to 255 and is available for Aztec Code Compact Format only.

11- **BarHeight :** Specifies the bar's height of the barcode. It's measured in inches. It is changing between 0.1 and 3. Only using with Code16k. Please see note at the end of 2D barcode object. It shows inch calculating.

12- **BarRatio :** Specifies the wide bar's width compared to the narrow bar's width. In barcode terminology this is N value. It is changing between 1 and 30. Please see note at the end of 2D barcode object. It shows inch calculating.

13- **BarWidth :** Specifies the narrow bar's width of the barcode. In barcode terminology this is X value. It's measured in inches. Please see note at the end of 2D barcode object. It shows inch calculating.

14- **BorderWidth :** Specifies the barcode image border's width. Border property is drawing frame around barcode. Default it is 0. If it is 0, cannot see any frame around barcode.

15- **Code16kMode :** Specifies the mode to use for Code16k symbology.It can be Mode0, Mode1 and Mode2.
  - a- **Mode0 :** This will use the Code 128 Char Set A which only supports ASCII values from 0 to 95. It is using as a **default**.
  - b- **Mode1 :** This will use the Code 128 Char Set B which only supports ASCII values from 32 to 127
  - c- **Mode2 :** This will use the Code 128 Char Set C which only supports pairs of digits

16- **Pdf417AspectRatio :** Specifiesthe ratio of the height to the overall width of the PDF417 symbol. Its value must be 0 (zero) up to 1 (one). **Default** it is 0.

17- **Pdf417Columns :** Specifiesthe number of columns to use for PDF417 symbology.

18- **Pdf417CompactionType :** Specifies the Compaction Type to apply for PDF417 symbology.
   a- **Auto :** It switches between Text, Binary and Numeric modes in order to minimize the number of codewords to be encoded.
   b- **Binary :** It allows encoding all 256 possible 8-bit byte values. This includes all ASCII characters value from 0 to 127 inclusive and provides for international character set support. It is using as a **Default**.
   c- **Text :** It allows encoding all printable ASCII characters, i.e. values from 32 to 126 inclusive in accordance with ISO/IEC 646, as well as selected control characters such as TAB (horizontal tab ASCII 9), LF (NL line feed, new line ASCII 10) and CR (carriage return ASCII 13).
   d- **Numeric :** It allows encoding numeric data strings.

19- **Pdf417Rows :** Specifies the number of rows to use for PDF417 symbology.

20- **Inverted :** If you can print with white ribbon you must select White Ribbon option. The barcode's color inverted automatic. This property is useful when the user prints dark colored pack

21- **CounterBegin :** Specifies counter start value for 2D barcode. This item is only used when Source item adjust as Counter.

22- **CounterEnd:** Specifies counter end value for 2D barcode. This item is only used when Source item adjust as Counter.

23- **CounterStep:** Specifies counters step value. Deafult it is 1. For 1, counter increases one by one. This item is only used when Source item adjust as Counter.

24- **CounterPeriod:** Specifies counter value increase after how many print. Deafult it is 1. This item is only used when Source item adjust as Counter.

25- **CounterDigit:** Specifies digit of counter for counter variable 2D barcode. This item is only used when Source item is Counter.

26- **FileName** : This item stores file name(if possible with path) which keeps data. File must be csv file. If file name will adjust in printer controller, it can be default.csv.This item is only used when Source item is database.

27- ColumnNo : Specifies column of datas in csv file. It is default 0.This item is only used when Source item is database.

Note : 2D Barcode component inch standart is 96 dpi but we are using 300 dpi(Print Head's resolution). So, you must be use this formula for calculating original size for savema printer

Orginal Size = Value*(300/96 dpi) and 1 inch=25,4 mm

For example : For 0.5 value equals 0.5 / (300/96 dpi)= 0.16 inch=4 mm

Example

```
~SPLTDS{<Template>
        <General>
                <MachineType>53x70I</MachineType>
                <Name>temp1_53.ronx</Name>
                <Width>640</Width>
                <Height>480</Height>
        </General>
        <Object>
                <ObjectType>2DBarcode</ObjectType>
                 <Name>2dbarcode1</Name>
                <X>0</X>
                <Y>0</Y>
                <W></W>
                <H></H>
                <Rotate>0</Rotate>
                <Hidden>False</Hidden>
                <Content>
                        <Source>Internal</Source>
            <TwoDBarcodeType>Datamatrix</TwoDBarcodeType>
           <TwoDBarcodeValue>123456789</TwoDBarcodeValue>
                         <ErrorCorrection>23</ErrorCorrection>
                        <CodeFormat>Auto</CodeFormat>
                        <ModuleSize>0.04</ModuleSize>
                        <Version>Auto</Version>
                        <Encoding>Auto</Encoding>
                        <AztecCodeRune>0</AztecCodeRune>
                        <BarHeight>0.4</BarHeight>
                        <BarRatio>0.01</BarRatio>
                       <BarWidth>3</BarWidth>
                       <BorderWidth>0</BorderWidth>
                       <SymbolSize>0.04</SymbolSize>
                       <Code16kMode>Mode0</Code16kMode>
```

	<Pdf417AspectRatio>0</Pdf417AspectRatio> <Pdf417Columns>5</Pdf417Columns> <Pdf417CompactionType>Binary</Pdf417CompactionType> <Pdf417Rows>0</Pdf417Rows> <Inverted>False</Inverted> <CounterBegin>000000</CounterBegin> <CounterEnd>999999</CounterEnd> <CounterStep>1</CounterStep> <CounterPeriod>1</CounterPeriod> <CounterDigit>6</CounterDigit> <FileName>c:\savema.csv</FileName> <ColumnNo>0</ColumnNo> </Content> <Font> <Name>Arial</Name> <Size>15</Size> <Style>Bold</Style> </Font> </Object> </Template>}^ --- This template has only one 2d barcode

3.1.3.11 Database

Database object is used for printing random data in each print. It gets data from data file(.csv) for print. Database object have four different properties which is shown in below,

1- **FileName** : This item stores file name(if possible with path) which keeps data. File must be csv file. If file name will adjust in printer controller, it can be default.csv.
2- **MaxCharNo**: : Specifies maximum usable count of characters which will use in Database Texts. Csv file contains values character count cannot higher than MaxCharNo value.
3- **ColumnNo** : Specifies column of datas in csv file. Default value is 0.
4- **Restart** : Specifies whether start again after database arrives end of file or not. It can be True or False.
    a. **True**: Database object starts again after arrive end of file.
    b. **False**: Database object waits on end of file and print last data.

Example	~SPLTDS{<Template> <General> <MachineType>53x70I</MachineType>

```
            <Name>temp1_53.ronx</Name>
            <Width>640</Width>
            <Height>480</Height>
        </General>
        <Object>
            <ObjectType>Database</ObjectType>
            <Name>database1</Name>
            <X>10</X>
            <Y>63</Y>
            <W>105</W>
            <H>33</H>
            <Rotate>180</Rotate>
            <Hidden>False</Hidden>
            <Content>
                <FileName>c:\savema.csv</FileName>
                <MaxCharNo>6</MaxCharNo>
                <ColumnNo>0</ColumnNo>
                <Restart>True</Restart>
            </Content>
            <Font>
                <Name>Arial</Name>
                <Size>20</Size>
                <Style>Bold,Italic</Style>
            </Font>
        </Object>
    </Template>}^   --- This template has only one richtext
```

3.1.3.12 Table

Table object is used for printing multi-row and multi-column datas in table format. Table object have two different properties which is shown in below,

1- TableData : This item stores table infos which is saved by PC. Cell properities and values are sperated with "^" charcther.

2- ImageData : This item storages Table image's string in Base64 standart. Images are must be convert to Base64 data for this parameter.

3-FirstColumnPerc: Specifies the percentage of the width compared to table width.

4-TableWidth: Specifies the table Width.

5- MaxTableHeight: Specifies the maximum table height.

6-Row Count: Specifies the row count of table without header.

NOTE: Table column number is 2. Cannot be decreased or increased. Other values rather than 2 is ignored. PC software cannot be edited this type of tables higher than 2 columns, but controller screen can be open the table with image data.

Example	~SPLTDS{<Template> <General> <MachineType>53x70I</MachineType> <Name>temp1_53.ronx</Name> <Width>600</Width> <Height>640</Height> </General> <Object> <ObjectType>Table</ObjectType> <Name>table1</Name> <X>10</X> <Y>63</Y> <W>600</W> <H>640</H> <Rotate>180</Rotate> <Hidden>False</Hidden> <Content> <FirstColumnPerc>%60</FirstColumnPerc> <TableWidth>600</TableWidth> <MaxTableHeight>640</MaxTableHeight> <RowCount>5</ RowCount > <TableData> header0~Header~Arial~20~Regular^tx1~1~Arial~20~Regular^t x2~2~Arial~20~Regular^tx3~3~Arial~20~Regular^tx4~4~Arial~ 20~Regular^tx5~5~Arial~20~Regular^tx6~6~Arial~20~Regular ^tx7~7~Arial~20~Regular^tx8~8~Arial~20~Regular^tx9~9~Aria l~20~Regular^tx10~10~Arial~20~Regular</TableData> <ImageData>iVBORw0KGgoAAAANSUhEUgAAAlgAAAEE CAYAAAD0wkrNAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv 8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAABK6SURBVHhe7 diBzts4rgbQff+X3jW3VOtRaVtxlHRUnwXOBcf8JBQmbmj8/9n+

	918AAKb6z3+3/8X/AQDgTT6wAAAm84EFADCZDywAgMl8YAE ********** Multiple Lines deleted ********** +15XyjrgMAwHy5gLey7rO2nG/UdQAAmC8X8FbWfdaW8426D gAA8+UC3sq6z9pyvlHXAQBgvlzAW1n3WVvON+o6AADMlwt4 K+s+a8v5Rl0HAID5cgFvZd1nbTnfqOsAADBfLuCtrPusLecbdR0A AObLBbyVdZ+15XyjrgMAwHy5gLey7rO2nG/UdQAAmC8X8Fb WfdaW8/0xZAAAZvjPf/8H9dAxj5UhnLgAAAAASUVORK5CYII=< /ImageData> </Content> <Font> <Name>Arial</Name> <Size>15</Size> <Style>Bold,Italic</Style> </Font> </Object> </Template>}^ --- This template has only one richtext

Example schema for this example is show below:

Header	
tx1	tx2
tx3	tx4
tx5	tx6
tx7	tx8
tx9	tx10

3.1.4 Font

This propery is used for adjusting object view and size. A lot of objects are using this property. These are;

a- Date
b- Time

c- Text
d- Counter
e- Barcode
f- 2D Barcode
g- Table
h- RichText(Font information come from Rtf Data)

Some objects do not use this property. These are;

a- Logo
b- Shape

Font property have three different items. These are ;

1- Name : Specifies name of font. The supported fonts are shown below: it can be add.
   a- Arial
   b- Courier New
   c- Gulim (for Korean language characters)
   d- Impact
   e- Simsun (For Chinese characters)
   f- SimHei (For Chinese characters)
   g- Tahoma
   h- Times New Roman
   i- Trebuchet MS
   j- Verdana
   k- Arabicfont (For Arabic characters)
   l- AMS_Arunalu (For Sinhalese characters)
   m- Sinhala – Kumudu (For Sinhalese characters)
   n- Radhika-PC (For Sinhalese characters)
   o- Sandaya (For Sinhalese characters)
   p- Sinhala InetFont (For Sinhalese characters)
   q- BNazanin (for Persian characters)

2- **Size :** Specifies font size of object. Font size unit is point. Default 20pt.

3- Style : Specifies style of text. It can be Regular, Bold, Italic or Bold,Italic.
   a- **Regular :** This is default font style and it is normal text.
   b- **Bold :** Text is shown as a bold.
   c- **Italic :** Text is shown as an italic.
   d- **Bold,Italic :** Text is shown bold and italic at the same time.

3.2 Load Template File from Printer

SPLLTF: Allows to load selected template file which is stored in printer. Template must be stored in printer otherwise printer doesn't load this template.
Printer sends OK message when loading template operation is successed or sends FAIL message when setting loading template operation is failed.

Using	**~SPLLTF{Template File Name}^** **Parameters;** **Template File Name :** Specifies template file name which will be loaded. **Note:** Savema template's extension name is ronx. Second numerical extension(before .ronx) is specified according to printer type. Can be _32 (for 32mm printers), _53 (for 53mm printers) and _107 . Return Value(On Successed) : ~ SPGRES{SPLLTF:OK}^ Return Value(On Failed) : ~ SPGRES{SPLLTF:FAIL}^
Example	~SPLLTF{temp1_53.ronx}^ -- temp1_53.ronx file loads in printer.(if printer has this template)

3.3 Get Active Template

SPLGAT : Returns active working template name from printer. This command doesn't have parameter.

Using	**~SPLGAT^**
Example	~SPLGAT^ Return Value(On Successed) : ~ SPGRES{SPLGAT:temp1_53.ronx}^ -- Printer sends active template name which name is temp1_53.ronx **Note:** Savema template's extension name is ronx. Second numerical extension(before .ronx) is specified according to printer type. Can be _32 (for 32mm printers), _53 (for 53mm printers) and _107 (107 mm printers).

3.4 Get Stored Templates

SPLGST : Returns all stored template file names from printer.This command doesn't have parameter.

Using	**~SPLGST^**
Example	~SPLGST^ Return Value(On Successed) : ~ SPGRES{SPLGST:temp1_53.ronx<abc_53.ronx<temp2_53.ronx}^ -- Printer sends all template names from 53mm printers in SPGRES command parameter. ~ SPGRES{ SPLGST:temp1_32.ronx<abc_32.ronx<temp2_32.ronx}^ -- Printer sends all template names from 32mm printers in SPGRES command parameter. **Note:** Savema template's extension name is ronx. Second numerical extension(before .ronx) is specified according to printer type. Can be _32 (for 32mm printers), _53 aand _107 (107 mm printers).

3.5 Create Data File

SPLCDF: Allows to create data file(.csv file) in printer . Needs two parameters. Printer sends OK message when creating data(.csv) file operation is successed or sends FAIL message when creating data(.csv) operation is failed.

Using	**~SPLCDF{Data File Name~gt~File Content}^** **Parameters;** **Data File Name :** Specifies data file name which will be stored in printer. **File Content :** This parameter must be arranged according to csv file rules. Datas must be ordered per row and if use more than one column per row, columns must be seperated with ~sc~ text. Return Value(On Successed) : ~ SPGRES{SPLCDF:OK}^ Return Value(On Failed) : ~ SPGRES{SPLCDF:FAIL}^
Example	**sample.csv is created with 3 rows and 1 column in below;** ~SPLCDF{sample.csv~gt~abc1 bce1 cde1}^ **sample.csv is created with 3 rows and 3 columns in below;** ~SPLCDF{sample.csv~gt~abc1~sc~abc2~sc~abc3 bce1~sc~bce2~sc~bce3 cde1~sc~cde2~sc~cde3}^

3.6 Get Stored Data Files

SPLGSD: Returns all stored data file names from printer.This command doesn't have parameter.

Using	**~SPLGSD^**
Example	~SPLGSD^ Return Value(On Successed) : ~ SPGRES{SPLGSD:abc.csv<datafile1.csv}^ -- Printer sends all data file names from printer in SPGRES command parameter.

3.7 Delete Template File

SPLDTF : This command deletes selected template file from printer. This command uses template file name as a parameter.

Printer sends OK message when deleting template file operation is successed or sends FAIL message when deleting templatefileoperation is failed.

Using	**~SPLDTF{Template File Name}^** **Parameters;** **Template File Name :** Specifiestemplate file name which will be deleted. **Note:** Savema template's extension name is ronx. Second numerical extension(before .ronx) is specified according to printer type. Can be _32 (for 32mm printers), _53 (for 53mm printers) and _107 (107 mm printers). Return Value(On Successed) : ~ SPGRES{SPLDTF:OK}^ Return Value(On Failed) : ~ SPGRES{SPLDTF:FAIL}^
Example	~SPLDTF{temp1_53.ronx}^ -- temp1_53.ronx file is deleted from printer.

3.8 Delete All Templates

SPLDTA : This command deletes all stored template file from printer.User must be carefull before use this command. Because printer deletes all template file after get this command. This command doesn't have parameter.

Printer sends OK message when deleting all template files operation is successed or sends FAIL message when deleting template filesoperation is failed.

Using	**~SPLDTA^** Return Value(On Successed) : ~ SPGRES{SPLDTA:OK}^ Return Value(On Failed) : ~ SPGRES{SPLDTA:FAIL}^
Example	~SPLDTA^ -- if printer has template file(s), all of them are deleted.

3.9 Delete Data File

SPLDDF : This command deletes selected data file from printer. This command uses data file name as a parameter.

Printer sends OK message when deleting data file operation is successed or sends FAIL message when deleting datafileoperation is failed.

Using	**~SPLDDF{Data File Name}^** **Parameters;** **Data File Name :** Specifies data file name which will be deleted. Return Value(On Successed) : ~ SPGRES{SPLDDF:OK}^ Return Value(On Failed) : ~ SPGRES{SPLDDF:FAIL}^
Example	~SPLDDF{datafile1.csv}^ -- datafile1.csv file is deleted from printer.

3.10 Delete All Data Files

SPLDDA : This command deletes all stored data file from printer.User must be carefull before use this command. Because printer deletes all data file after get this command. This command doesn't have parameter.

Printer sends OK message when deleting all data files operation is successed or sends FAIL message when deleting all datafilesoperation is failed.

Using	**~SPLDDA^** Return Value(On Successed) : ~ SPGRES{SPLDDA:OK}^ Return Value(On Failed) : ~ SPGRES{SPLDDA:FAIL}^

3.11 Clear Data Buffer

SPLCDB : This command clears buffer which stored database datas as temporarily. When load template which have CSV database field, CSV datas and index of data(for start print) are loaded to data buffer. When delete CSV file, buffer should be cleared. This command doesn't have parameter.
Printer sends OK message when send this command to printer.

Using	**~SPLCDB^** Return Value: ~ SPGRES{SPLCDB:OK}^

3.12 Load Font File

SPLLFF: Allows to load font file(.ttf file) into printer . Needs two parameters.
Printer sends OK message when loading font file operation is successed or sends FAIL message when loading font file operation is failed.

Using	**~SPLLFF{Font File Name>File Content}^** **Parameters;** **Font File Name :** Specifies font file name which will be sent to printer. **File Content :** This parameters data must be read in binary format (as a byte array) after that converted to base64 format. Return Value(On Successed) : ~ SPGRES{SPLLFF:OK}^ Return Value(On Failed) : ~ SPGRES{SPLLFF:FAIL}^
Example	~SPLLFF{CENTURY.TTF>AAEAAAATAQAABAAwRFNJR/iDHXwAAmsAAAAagEx UU0i7RI/wAAAMiAAAAqJPUy8ydN5tGgAAAbgAAABWVkRNWAgm1vUAAA8s AAAXbmNtYXDa6Gk6AABlpAAABlJjdnQgrT+zvwAAdrAAAAUcZnBnbe485joAA Gv4AAAEgWdhc3AAGQAJAAJq8AAAABBnbHlmSH/pt ********** Multiple Lines deleted ********** pABD8btUIlP05gL/LxaNIb8AEOfPwhuC4MnD4alsNWb04K1ZrSwmOK9TC6Z2+ 1+Wm82BPIl5EipYMq/BTYIoL44U/tlak8PNztIEpeK/+IRNJLhuG2Lh3Me4RdPK Od8ZR7VujwwWfif/tKMvTkN3qa5jAsBMKtg3ZLOkBGqGWBVc9tBMycSJtG3H VE5dXRHPcfIDwP3hPp1xuGzjFlE+m6wmtOzvs/0v67QzxE14IGXtbAGFtAoz9d WQFFReySQlbIfI+oz+QoEaB0rASm17wwncAA=}^

3.13 Get Font Files

SPLGFF: Returns all loaded font file names from printer.This command doesn't have parameter.

Using	**~SPLGFF^**
Example	~SPLGFF^ Return Value(On Successed) : ~ SPGRES{SPLGFF:arial.ttf<tahoma.ttf<verdana.ttf}^ -- Printer sends all font file names from printer in SPGRES command parameter.

3.14 Delete Font File

SPLDFF : This command deletes selected font file from printer. This command uses font file name as a parameter.
Printer sends OK message when deleting font file operation is successed or sends FAIL message when deleting font file operation is failed.

Using	**~SPLDFF{Font File Name}^** **Parameters;** **Font File Name :** Specifies font file name which will be deleted.If printer doesn't have specified font file, command returns FAIL message. **Note**: Printer must be restart after delete any fonts. Return Value(On Successed) : ~ SPGRES{ SPLDFF:OK}^ Return Value(On Failed) : ~ SPGRES{ SPLDFF:FAIL}^
Example	~SPLDFF{arial.ttf}^ -- arial.ttf file is deleted from printer.

3.15 Get Field Names

SPLGFN : This command returns field names of selected template file which is stored in printer. This command uses template file name as a parameter.
Printer sends field names with template name when get this command. .If printer doesn't have specified template file, command returns FAIL message.

Using	**~SPLGFN{Template File Name}^** **Parameters;** **Template File Name :** Specifies template file name which is stored in printer. Return Value(On Successed) : ~ SPGRES{SPLGFN:template name<field name 1<field name 2<…}^ Return Value(On Failed) : ~ SPGRES{SPLGFN:FAIL}^
Example	~SPLDFF{temp1_53.rox}^ -- Returns fields name of temp1_53.rox template. ~ SPGRES{SPLGFN: temp1_53.rox <Prod. Name<Prod. Dat<Exp.Date}^

4. Modification Commands

Modification commands allows to chnage Text, Barcode and 2D barcode in a template. This commands are generally used to change one or more objects value at the same time in template over Ethernet communication. This commands can use one by one or with together.
Source option of changable object(Text, Barcode and 2D barcode) must be **External** for modification.
For external objects, data value must be carefullly checked.
In template rotation 0 or 180 degrees, topmost or bottommost external object's height and width carefully controlled. If control's width/height exceeds template width/height, some clipping may occur.
In template rotation 90 or 270 degrees, leftmost or rightmost external object's height and width carefully controlled. If control's width/height exceeds template width/height, some clipping may occur.

Note : In order to these commands operating as intended and working properly, commands must be sended either machine is stop position or alternatively machine is print position and package is stop position.

4.1 Changing Text Value Commands

SPMCTV: This command changes selected Text object value in template. Related text object must be in template. Otherwise commands doesn't change any text. Printer sends OK message when changing text value operation is successed or sends FAIL message when changing tex valueoperation is failed.

Using	**~SPMCTV{Name of object~gt~Text Value}^** **Parameters;** **Name of object:** Please entere name of text object which is defined in PC software. **Text Value :**New value of selected text. Return Value(On Successed) : ~ SPGRES{SPMCTV:OK}^ Return Value(On Failed) : ~ SPGRES{SPMCTV:FAIL}^
Example	~SPMCTV{brand_txt~gt~SAVEMA}^ -- Set brand_txt value to "SAVEMA" ~SPMCTV{type_txt~gt~PRINTER}^ -- Set type_txt value to "PRINTER"

4.2 Changing Barcode Value Commands

SPMCBV : This command changes selected Barcode(1D) object value in template. Related barcode object must be in template. Otherwise commands doesn't change any barcode.

Printer sends OK message when changing barcode value operation is successed or sends FAIL message when changing barcode valueoperation is failed.

Using	**~SPMCBV{Name of object**~gt~**Barcode Value}^** **Parameters;** **Name of object:** Please entere name of barcode object which is defined in PC software. **Barcode Value :**New value of selected barcode. Barcode value must be compatible with barcode type. Forexample EAN-13 barcode type accept only numerical characters and this value must be compatible with EAN-13 rules. Barcode value characters count must be compatible with barcode type. Forexample, EAN-13 barcode accepts 12 or 13 numerical characters. Return Value(On Successed) : ~ SPGRES{SPMCBV:OK}^ Return Value(On Failed) : ~ SPGRES{SPMCBV:FAIL}^
Example	~SPMCBV{bar1~gt~8691234567890}^ -- Set bar1 value to 8691234567890

4.3 Changing 2D Barcode Value Commands

SPMC2D : This command changes selected 2D Barcode object value in template. Related 2D barcode object must be in template. Otherwise commands doesn't change any 2D barcode.

Printer sends OK message when changing 2D barcode value operation is successed or sends FAIL message when changing 2D barcode valueoperation is failed.

Using	**~SPMC2D{Name of object**~gt~**Barcode Value}^** **Parameters;** **Name of object:** Please entere name of 2D barcode object which is defined in PC software.

	Barcode Value : New value of selected 2D barcode. Barcode value must be compatible with barcode type. Forexample Datamatrix barcode accepts only standart ASCII characters but cannot accepts extra character without tandart ascii character. Forexample Datamatrix barcode type doesn't accepts Ç,ü,ş,ö characters or arabic letters or chinese letters. Standart ASCII characters; *!"#$%&'()*+,-* *./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_abcdefghij* *klmnopqrstuvwxyz{|}~* Return Value(On Successed) : ~ SPGRES{SPMC2D:OK}^ Return Value(On Failed) : ~ SPGRES{SPMC2D:FAIL}^
Example	~SPMC2D{b2bar1~gt~savema12345}^ -- Set b2bar1 value to savema12345

4.4 Changing Counter Value Commands

SPMCCV : This command changes selected Counter object value in template. Related counter object must be in template. Otherwise commands doesn't change any counter.
Printer sends OK message when changing counter value operation is successed or sends FAIL message when changing counter value operation is failed.

Using	**~SPMCCV{Name of object**~gt~**Counter Value}^** **Parameters;** **Name of object:** Please entere name of counter object which is defined in PC software. **Counter Value :**New value of selected counter. Counter value must be numeric. Return Value(On Successed) : ~ SPGRES{SPMCCV:OK}^ Return Value(On Failed) : ~ SPGRES{SPMCCV:FAIL}^
Example	~SPMCCV{counter1~gt~000055}^ -- Set counter1 value to 000055

4.5 Changing Selected Values Command

SPMCSV: This command changes selected Text, barcode and 2d barcode object's value in template. Related object must be in template. Otherwise commands doesn't change any object. You can modificate one or more objects value. Printer sends OK message when changing text value operation is successed or sends FAIL message when changing text valueoperation is failed.

Using	**~SPMCSV{Name of object**~gt~**Value**~gt~**Name of object**~gt~**Value}^** **Parameters;** **Name of object:** Please entere name of text object which is defined in PC software. **Text Value :**New value of selected text. Return Value(On Successed) : ~ SPGRES{SPMCSV:OK}^ Return Value(On Failed) : ~ SPGRES{SPMCSV:FAIL}^
Example	~SPMCTV{brand_txt~gt~SAVEMA~gt~barcodeno~gt~123456789125}^ -- Set brand_txt value to "SAVEMA" and barcode value as 123456789125

5. Print Commands

5.1 Start Print

SPPSAP : This command starts printing automatically. Printer must be ready before send this command. Otherwise printer cannot start automatic printing. This command doesn't have parameter.
Printer sends OK message when starting print operation is successed or sends FAIL message when starting printoperation is failed.

Using	**~SPPSAP^** Return Value(On Successed) : ~ SPGRES{SPPSAP:OK}^ Return Value(On Failed) : ~ SPGRES{SPPSAP:FAIL}^
Example	~SPPSAP^ -- if printer isready, printer starts printing automatically.

5.2 Set/Get Print Count for Limited print

SPPSLQ : This commad specifies print quantity for limited prints. SPPSAP command must be send end of this command for start printing. Otherwise printer doesn't start to print. Printer will stop after print quantity arrive to 0. This value is automatically decreasing one by one. This command have one parameter.
Printer sends OK message when setting print quantity operation is successed or sends FAIL message when setting print quantityoperation is failed.

Using	**~SPPSLQ{Limited Print Quantity}^** **Parameters;** **Limited Print Quantity:**Provides controlled printing. Printer works until print quantity is 0. Return Value(On Successed) : ~ SPGRES{SPPSLQ:OK}^ Return Value(On Failed) : ~ SPGRES{SPPSLQ:FAIL}^
Example	~SPPSLQ{1000}^ -- Limited Print Quantity is specified as 1000. Printer doesn't start printing. ~SPPSLQ{1000}|SPPSAP^-- Printer prints 1000 prints and stop.

SPPGLQ : This command retuns actual print quantity value. This command doesn't have parameter.

Using	**~SPCGLQ^**
Example	~SPCGLQ^ Return Value(On Successed) : ~ SPGRES{SPCGLQ:500}^ -- Print quantity is 500.

5.3 Stop Print

SPPSTP : This command stops printing . Printer must be working before send this command. Otherwise printer cannot stop printing. This command doesn't have parameter.

Printer sends OK message when stop print operation is successed or sends FAIL message when stop printoperation is failed.

Using	**~SPPSTP^** Return Value(On Successed) : ~ SPGRES{SPPSTP:OK}^ Return Value(On Failed) : ~ SPGRES{SPPSTP:FAIL}^
Example	~SPPSTP^ -- if printer is working , this command stops printing.

5.4 One Test Print

SPPOTP : Provides to print one time . This command doesn't have parameter.

Printer sends OK message when print is successed or sends FAIL message when print is failed.

Using	**~SPPOTP^** Return Value(On Successed) : ~ SPGRES{SPPOTP:OK}^ Return Value(On Failed) : ~ SPGRES{SPPOTP:FAIL}^
Example	~SPPOTP^ -- Printer prints one time.

5.5 Status of Printer

SPPSTA : This command returns status of printer. There is 4 different response. These are INIT, WAITING, RUNNING and ERROR.

1- **INIT** : Printer sends when controller software is loading in startup.(Before loaded template automatically). When template loaded automatically, WAITING message is sent by printer.
2- **WAITING** : Printer sends when printer in **stop** mode.(Stop button pressed)
3- **RUNNING** : Printer sends when printer in **printing** mode.(Print button pressed)
4- **ERROR**: Printer sends when any error happens in printer. Printer sends error type in response message.

This command doesn't have parameter.

Note: Printer sends FAIL message for all commands(except SPPSTA) when operator doesn't in main window. SPPSTA command sends BLOCKED message with printer status.

Using	**~SPPSTA^** Return Value(in startup- before loaded template) : ~ SPGRES{SPPSTA:INIT<}^ Return Value(in stop mode) : ~ SPGRES{SPPSTA:WAITING<}^ Return Value(in running mode) : ~ SPGRES{SPPSTA:RUNNING<}^ Return Value(when error happens) : ~ SPGRES{SPPSTA:ERROR<Error Content}^ **When operator doesn't in main window;** Return Value(in startup- before loaded template) : ~ SPGRES{SPPSTA:INIT<BLOCKED}^ Return Value(in stop mode) : ~ SPGRES{SPPSTA:WAITING<BLOCKED}^ Return Value(in running mode) : ~ SPGRES{SPPSTA:RUNNING<BLOCKED}^ Return Value(when error happens) : ~ SPGRES{SPPSTA:ERROR<BLOCKED Error Content}^
Example	~SPPSTA^ Return in error mode; ~ SPGRES{SPPSTA:ERROR< Ribbon not found.Please insert ribbon}^ **When operator doesn't in main window;** ~ SPGRES{SPPSTA:ERROR<BLOCKED Ribbon not found.Please insert ribbon}^

6.General Commands

6.1 Send User Message to Printer

SPGSUM : This command provides to show coming message to the printer display. Message can be received from PC, PLC or another device which sends this command. This command has one parameter.

Using	**~SPGSUM{User Message}^** **Parameters;** **User Message :**Sent from connected device with printer and showed on printer display. This command is used for warning purposes. So, it doesn't affect printer.
Example	~SPGSUM{Package finished. Please stop printer}^ -- Printer has received "*Package finished. Please stop printer"* message from pack machine.

6.2 General Response From Printer

SPGRES : Returns all response from printer when request command is processed. So, this command cannot be used directly, only printer gives sends this command to connected device.

This command have one parameter and this parameter content changes according to request command.

Using	**~SPGRES{Response}^** **Parameters;** **Response :** This parameters content changes according to request command.
Example	~SPGRES{SPGDTP:950225}^ -- Returns total print count ~SPGRES{SPCGPA:25<27<300<200<31<77<0<24<25<0<1265<0<5<0<23<0<4<0<0<400}^ --Retuns all sistem parameter..

6.3 Get Total Print Count

SPGGTP : Returns total print count of printer.This command doesn't have parameter.

Using	**~SPGGTP^**
Example	~SPGGTP^ Return Value(On Successed) : ~ SPGRES{SPGGTP:458200}^ -- Printer printed 458200 prints since it started working first.

6.4 Get Current Print Count

SPGGCP : Returns current print count of printer.This counter resets when load any template. This count is shown in main window. This command doesn't have parameter.

Using	**~SPGGCP^**
Example	~SPGGCP^ Return Value(On Successed) : ~ SPGRES{SPGGTP:1250}^ -- Printer printed 1250 prints since load template.

6.5 Get Firmware Version

SPGGFV : Returns firmware version of printer. This command doesn't have parameter.

Using	**~SPGGFV^** Note:
Example	~SPGGFV^ Return Value(On Successed) : ~ SPGRES{SPGGFV:6.3.001.600.R}^ -- Printer firmware version is 6.3.001.600.R

6.6 Get Remaining Ribbon(for printers models with Cassette)

SPGGRR : Returns remaining ribbon percentage. This command is used with printer models with cassette.This command doesn't have parameter.

Using	**~SPGGRR^** Note:
Example	~SPGGRR^ Return Value(On Successed) : ~ SPGRES{SPGGRR:80}^ -- Remaining ribbon amount is 80%.

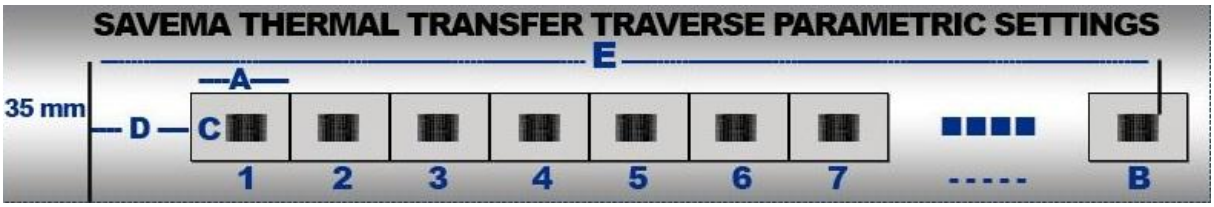
6.7 Get Serial Number of Printer

SPGGSN : Returns serial number of printer.This command doesn't have parameter.

Using	**~SPGGSN^**
Example	~SPGGSN^ Return Value(On Successed) : ~ SPGRES{SPGGSN:17013012}^ -- Printer serial number is 17013012.

7.TraverseCommands

Traverse Commands are using onlyin traverse printers(TR53 and TR107). Traverse printers allows to print on multi-packages with one print signal. Traverse printer have some parameters and they are specified in below scheme.



A- Pack Size(mm)
B- Print Count(mm)
C- Print Position in one package(mm)
D- Package Distance from beginnig of package
E- Printing area

7.1 Set/Get Pack Size(A)

SPTSPS: Allows to set one package size in multi-package.This value is mesaured with millimeter.

Using	**~SPTSPS{Pack size}^** **Parameters;** **Pack size:**Specifies one partial package size in millimeter.Value must be between 1-3000. Return Value(On Successed) : ~ SPGRES{ SPTSPS:OK}^ Return Value(On Failed) : ~ SPGRES{ SPTSPS:FAIL}^
Example	~SPTSPS{60}^

SPTGPS : Returnsone package width in multi-package.

Using	**~SPTGPS^**
Example	~SPTGPS^ Return Value(On Successed) : ~ SPGRES{ SPTGPS:60}^

7.2 Set/Get Print Count(B)

SPTSPC: Allows to set print count in one print signal. Print count must be specified according to package count in multi-package.

Using	**~SPTSPC{Print Count}^** **Parameters;** **Print Count:** Specifies print count in one print signal Value must be between 1-3000. Return Value(On Successed) : ~ SPGRES{ SPTSPC:OK}^ Return Value(On Failed) : ~ SPGRES{ SPTSPC:FAIL}^
Example	~SPTSPC{5}^

SPTGPC : Returnsprint count.

Using	**~SPTGPC^**
Example	~SPTGPC^ Return Value(On Successed) : ~ SPGRES{ SPTGPC:5}^

7.3 Set/Get Print Position(C)

SPTSPP: Allows to set print position of template from beginning of one package .This value is mesaured with millimeter.

Using	**~SPTSPP{Print Position}^** **Parameters;** **Print Position:** Specifies print position of template. Value must be between 1-3000. Return Value(On Successed) : ~ SPGRES{ SPTSPP:OK}^ Return Value(On Failed) : ~ SPGRES{ SPTSPP:FAIL}^
Example	~SPTSPP{10}^

SPTGPP: Returns print position of template in one package.

Using	**~SPTGPP^**
Example	~SPTGPP^ Return Value(On Successed) : ~ SPGRES{ SPTGPP:10}^

7.4 Set/Get Package Distance(D)

SPTSPD: Allows to set package distance from beginnig of printer printing area. This value is mesaured with millimeter.

Using	**~SPTSPD{Pack Distance}^** **Parameters;** **Pack Distance:** Specifies package distance from beginning of printing area. Value must be between 0-3000. Return Value(On Successed) : ~ SPGRES{ SPTSPD:OK}^ Return Value(On Failed) : ~ SPGRES{ SPTSPD:FAIL}^
Example	~SPTSPD{50}^

SPTGPD: Returns package distance from beginning of printing area.

Using	**~SPTGPD^**
Example	~SPTGPD^ Return Value(On Successed) : ~ SPGRES{ SPTGPD:50}^

7.5 Set/Get Printing Area(E)

SPTSPA: Allows to set total printing area of traverse printer. This value must be specified according to traverse printer characteristics(max. printing area). This value is mesaured with millimeter.

Using	**~SPTSPA{Printing Area}^** **Parameters;** **Printing Area:** Specifies printing area of traverse printer. Value must be between 0-3000. Return Value(On Successed) : ~ SPGRES{ SPTSPA:OK}^ Return Value(On Failed) : ~ SPGRES{ SPTSPA:FAIL}^
Example	~SPTSPA{400}^

SPTGPA: Returns printing area of traverse printer.

Using	**~SPTGPA^**
Example	~SPTGPA^ Return Value(On Successed) : ~ SPGRES{ SPTGPA:400}^

7.6 Set/Get All Traverse Parameters

SPTSTP: Allows to set all traverse parameters for traverse printer.

Using	**~SPTSTP{Pack Size>Print Count>Print Position>Pack Distance>Printing Area}^** **Parameters;** **Pack size:** Specifies one partial package size in millimeter.Value must be between 1-3000. **Print Count:** Specifies print count in one print signal. Value must be between 1-3000. **Print Position:** Specifies print position of template. Value must be between 1-3000. **Pack Distance:** Specifies package distance from beginning of printing

	area. Value must be between 0-3000. **Printing Area:** Specifies printing area of traverse printer. Value must be between 0-3000. Return Value(On Successed) : ~ SPGRES{SPTSTP:OK}^ Return Value(On Failed) : ~ SPGRES{SPTSTP:FAIL}^
Example	~SPTSTP{60>5>10>50>400}^

SPTGTP: Returnsall traverse parameters.

Using	**~SPTGTP^**
Example	~SPTGTP^ Return Value(On Successed) : ~ SPGRES{ SPTGTP:60<5<10<50<400}^

8.System Parameters Explanation

Parameter No	32x40/50 I & 53x40/50 I	Min.- Max. Value
P1	Motor ramp starting value (0-150)	0-150
P2	TPH down time (0-150)	0-150
P3	Ribbon and termal mechanic rewind speed (100-500)	100-500
P4	Ribbon space (0-1000) [Ribbon space decrease < 200] [Ribbon space increase > 200]	0-1000
P5	Before the suspension point of the diameter of the surface of the ribbon hanging (10-60) [This value is constant for our ribbon winding system on printer] [Effect : Writing speed calculate system] [Factory Setting]	10-60
P6	After the suspension point of the diameter of the surface of the ribbon hanging (0-100) [This value is constant for our ribbon winding system on printer] [Effect : Writing speed calculate system] [Factory Setting]	0-100
P7	Vertical printing wait. when machine set up vetical, horizontal motor moment of inertia waiting (0-3000)	0-3000
P8	Pre-Heating system control temperature as celsius (24-40) [This value determine minimum TPH temperature]	24-40
P9	Before the suspension point of the diameter of the surface of the pulley (10-60) [Effect : Writing speed calculate system] [Factory Setting]	10-60
P10	After the suspension point of the diameter of the surface of the pulley (0-100) [Effect : Writing speed calculate system] [Factory Setting]	0-100
P11	TPH resistance value. (600-2000)	600-2000

P12	Change with 20. Parametric . Difficult material print. (0-2) [0: Easy mod. Max. Speed must be up to 500 (20. Parametic and on setting menu-> print speed] [1: Pre-difficult mod. Max. print Speed must be up to 300 (20. Parametic and on setting menu-> print speed] [2: Difficult mod. Max.print Speed must be up to 200 (20. Parametic and on setting menu-> print speed]	0-2
P13	Contact signal waiting value. (0-1000)	0-1000
P14	Ribbon break active or inactive (0-1) [0: Ribbon break passive] [1: Ribbon break active]	0-1
P15	Print head printing stop position. (0-400) [If the last out of the template does not transfered, value is increased.]	0-400
P16	Pre-heating system active or inactive selection (0-1) [0: Pre-heating system active] [1: Pre-heating system inactive]	0-1
P17	Ribbon break limitation (0-1000)	0-1000
P18	Printer fuse error warning active or inactive setting (0-3) [0: Main board fuse warning ACTIVE, Motor driver fuse warning ACTIVE] [1: Main board fuse warning INACTIVE, Motor driver fuse warning ACTIVE] [2: Main board fuse warning ACTIVE, Motor driver fuse warning INACTIVE] [3: Main board fuse warning INACTIVE, Motor driver fuse warning INACTIVE]	0-3
P19	Fault output mode (0-1) [0: Machine gives alarms ERROR times and inactive times] [1: Machine gives alarm ERROR times]	0-1
P20	Max Print speed (150-500)	150-500

Table-2) 32x40/50I and 53x40/50 I Parameters Explanation

Parameter No	32x70 I & 53x70/125 I &107x75/125 I	Min.- Max. Value
P1	Motor ramp starting value (0-150)	0-150
P2	TPH down time (0-150)	0-150
P3	Ribbon and termal mechanic rewind speed (100-500)	100-500
P4	Ribbon space (0-1000) [Ribbon space decrease < 200] [Ribbon space increase > 200]	0-1000
P5	Before the suspension point of the diameter of the surface of the ribbon hanging (10-60) [This value is constant for our ribbon winding system on printer] [Effect : Writing speed calculate system] [Factory Setting]	10-60
P6	After the suspension point of the diameter of the surface of the ribbon hanging (0-100) [This value is constant for our ribbon winding system on printer] [Effect : Writing speed calculate system] [Factory Setting]	0-100
P7	Vertical printing wait. when machine set up vertical, horizontal motor moment of inertia waiting (0-3000)	0-3000
P8	Pre-Heating system control temperature as celsius (24-40) [This value determine minimum TPH temperature]	24-40
P9	Before the suspension point of the diameter of the surface of the pulley (10-60) [Effect : Writing speed calculate system] [Factory Setting]	10-60
P10	After the suspension point of the diameter of the surface of the pulley (0-100) [Effect : Writing speed calculate system] [Factory Setting]	0-100
P11	TPH resistance value. (600-2000)	600-2000
P12	Change with 20. Parametric . Difficult material print. (0-2) [0: Easy mod. Max. Speed must be up to 500 (20. Parametic and on setting menu-> print speed] [1: Pre-difficult mod. Max. print Speed must be up to 300 (20. Parametic and on setting menu-> print speed] [2: Difficult mod. Max.print Speed must be up to 200 (20. Parametic and on setting menu-> print speed]	0-2
P13	Contact signal waiting value. (0-1000)	0-1000

P14	Ribbon break and Ribbon not found active or passive (0-3) [0: Ribbon break active,Ribbon not found active] [1: Ribbon break passive,Ribbon not found active] [2: Ribbon break passive,Ribbon not found passive] [3: Ribbon break and Ribbon not found are in switch mode]	0-3
P15	Print head printing stop position. (0-400) [If the last out of the template does not transfered, value is increased.]	0-400
P16	Pre-heating system active or inactive selection (0-1) [0: pre-heating system active] [1:pre-heating system inactive]	0-1
P17	Ribbon break limitation (0-1000)	0-1000
P18	Printer fuse error warning active or inactive setting (0-3) [0: Main board fuse warning ACTIVE, Motor driver fuse warning ACTIVE] [1: Main board fuse warning INACTIVE, Motor driver fuse warning ACTIVE] [2: Main board fuse warning ACTIVE, Motor driver fuse warning INACTIVE] [3: Main board fuse warning INACTIVE, Motor driver fuse warning INACTIVE]	0-3
P19	Fault output mode (0-1) [0: Machine gives alarms ERROR times and inactive times] [1: Machine gives alarm ERROR times]	0-1
P20	Max Print speed (150-500)	150-500

Table-3) 32x70I and 53x70/125I and 107x75/125 I Parameters Explanation

Parameter No	32C	Min.- Max. Value
P1	Measuring the diameter of the encoder as milimeter (10-70) [Effect : Substrate speed detection system] [Factory setting]	10-70
P2	Pulse number of the encoder (800-1500) [Factory setting]	80-1500
P3	Diameter of red roller (20-60) [This value is constant for our ribbon winding system on printer] [Effect : Writing speed calculate system] [Factory setting]	20-60
P4	Diameter of red roller decimal portion value (0-100) [This value is constant for our ribbon winding system on printer] [effect : writing speed calculate system] [Factory setting]	0-100
P5	Reserved (0) [Factory setting]	0
P6	TPH Hold(pressed) encoder pulse number (0-600)	0-600
P7	Motor ramp distance as milimeter (0-70) [Factory setting]	0-70
P8	TPH mechanism down time as milisecond (0-50) [Factory setting]	0-50
P9	Ribbon space decrease (0-1000) [This value decrease the gap of two printout on ribbon.] [Effect: Each 10 motor pulse effect 1mm]	0-1000
P10	Ribbon space increase (0-1000) [This value increase the gap of two printout on ribbon.] [effect:each 10 motor pulse effect 1mm]	0-1000
P11	TPH resistance value (600-2000)	600-2000
P12	Ribbon Settings [0 : Wax-resin + Easy package, Speed to 600] [1 : Wax-resin + Difficult package or Resin + Easy Package, Speed to 550] [2 : Wax-resin + Very difficult package or Resin + Difficult package, Speed to 400]	0-2

P13	Continous prints modes (0-5) [0 : Working speed 30-300 mm/sc, min. pack size=50mm+ Template Size] [1 : Working speed 30-350 mm/sc, min. pack size=50mm+ Template Size] [2: Working speed 30-400 mm/sc, min. pack size=55mm+ Template Size] [3: Working speed 30-450 mm/sc, min. pack size=60mm+ Template Size] [4: Working speed 30-550 mm/sc, min. pack size=65mm+ Template Size] [5: Working speed 30-600 mm/sc, min. pack size=70mm+ Template Size]	0-5
P14	Conservation value of the minimum speed as mm/sc. (0-1000) [This value is cancel limit when band speed less than this value, while printer printing]	0-1000
P15	Pre-Heating system control temperature as celsius (24-40) [This value determine minimum TPH temperature]	24-40
P16	Pre-heating system active or inactive selection (0-1) [0: Pre-heating system active] [1: Pre-heating system inactive]	0-1
P17	Reserved (0) [Factory setting]	0
P18	Printer fuse error warning active or inactive setting (0-3) [0: Main board fuse warning ACTIVE, Motor driver fuse warning ACTIVE] [1: Main board fuse warning INACTIVE, Motor driver fuse warning ACTIVE] [2: Main board fuse warning ACTIVE, Motor driver fuse warning INACTIVE] [3: Main board fuse warning INACTIVE, Motor driver fuse warning INACTIVE]	0-3
P19	Fault output mode (0-1) [0: Machine gives alarms ERROR times and inactive times] [1: Machine gives alarm ERROR times]	0-1
P20	Determine the direction of ENCODER (0-1) [0: anti-clock wise rotation] [1: clock wise rotation]	0-1

Table-4) 32C Parameters Explanation

Parameter No	32 CC & 53 C &107 C	Min.- Max. Value
P1	Measuring the diameter of the encoder as milimeter (10-70) [Effect : Substrate speed detection system] [Factory setting]	10-70
P2	Pulse number of the encoder (80-1500) [Factory setting]	80-1500
P3	Diameter of red roller (20-60) [This value is constant for our ribbon winding system on printer] [Effect : Writing speed calculate system] [Factory setting]	20-60
P4	Diameter of red roller decimal portion value (0-100) [This value is constant for our ribbon winding system on printer] [Effect : Writing speed calculate system] [Factory setting]	0-100
P5	Reserved (0) [Factory setting]	0
P6	TPH Hold(pressed) encoder pulse number (0-600)	0-600
P7	Ribbon break and Ribbon not found active or passive (0-3) [0: Ribbon break active,Ribbon not found active] [1: Ribbon break passive,Ribbon not found active] [2: Ribbon break passive,Ribbon not found passive] [3: Ribbon break and Ribbon not found are in switch mode]	0-3
P8	TPH mechanism down time as milisecond (0-50) [Factory setting]	0-50
P9	Ribbon space decrease (0-1000) [This value decrease the gap of two printout on ribbon.] [Effect: Each 10 motor pulse effect 1mm]	0-1000
P10	Ribbon space increase (0-1000) [This value increase the gap of two printout on ribbon.] [Effect: Each 10 motor pulse effect 1mm]	0-1000
P11	TPH resistance value. (600-2000)	600-2000
P12	Ribbon Settings [0 : Wax-resin + Easy package, Speed to 800(32CC,53C), 600(107C)] [1 : Wax-resin + Difficult package or Resin + Easy Package, Speed to 550] [2 : Wax-resin + Very difficult package or Resin + Difficult package, Speed to 400]	0-2

P13	Continous prints modes (0-5) [0 : Working speed 30-400 mm/sc, min. pack size=50mm+ Template Size] [1 : Working speed 30-450 mm/sc, min. pack size=50mm+ Template Size] [2: Working speed 30-500 mm/sc, min. pack size=55mm+ Template Size] [3: Working speed 30-650 mm/sc, min. pack size=60mm+ Template Size] [4: Working speed 30-750 mm/sc, min. pack size=65mm+ Template Size] [5: Working speed 30-800 mm/sc, min. pack size=75mm+ Template Size]	0-5
P14	Conservation value of the minimum speed as mm/sc. (0-1000) [This value is cancel limit when band speed less than this value, while printer printing]	0-1000
P15	Pre-Heating system control temperature as celsius (24-40) [This value determine minimum TPH temperature]	24-40
P16	Pre-heating system active or inactive selection (0-1) [0: Pre-heating system active] [1: Pre-heating system inactive]	0-1
P17	Ribbon break limitation (0-1000)	0-1000
P18	Printer fuse error warning active or inactive setting (0-3) [0: Main board fuse warning ACTIVE, Motor driver fuse warning ACTIVE] [1: Main board fuse warning INACTIVE, Motor driver fuse warning ACTIVE] [2: Main board fuse warning ACTIVE, Motor driver fuse warning INACTIVE] [3: Main board fuse warning INACTIVE, Motor driver fuse warning INACTIVE]	0-3
P19	Fault output mode (0-1) [0: Machine gives alarms ERROR times and inactive times] [1: Machine gives alarm ERROR times]	0-1
P20	Determine the direction of ENCODER (0-1) [0: Anti-clock wise rotation] [1: Clock wise rotation]	0-1

Table-5) 32C with Cassette and 53C and 107C I Parameters Explanation

Parameter No	32TR & 53TR & 107TR	Min.- Max. Value
P1	Motor ramp starting value (0-30)	0-30
P2	TPH down time (0-30)	0-30
P3	Ribbon and thermal mechanic rewind speed (150-700)	150-700
P4	Ribbon space (50-1000) [Ribbon space decrease < 200] [Ribbon space increase > 200]	50-1000
P5	Before the suspension point of the diameter of the surface of the ribbon hanging (10-60) [This value is constant for our ribbon winding system on printer] [Factory Setting]	10-60
P6	After the suspension point of the diameter of the surface of the ribbon hanging (0-99) [This value is constant for our ribbon winding system on printer] [Factory Setting]	0-99
P7	Ribbon reduction mode value (0-600)	0-600
P8	Pre-Heating system control temperature as celsius (24-45) [This value determine minimum TPH temperature]	24-45
P9	TPH Pressed minimum value for less than 300 mm/sec.(0-50)	0-50
P10	TPH Pressed minimum value for more than 300 mm/sc. (0-50)	0-50
P11	Printing point constant(0-30)	0-30
P12	Change with 20. Parametric . Difficult material print. (0-2) [0: Easy mod. Max. Speed must be up to 500 (20. Parametic and on setting menu-> print speed] [1: Pre-difficult mod. Max. print Speed must be up to 500 (20. Parametic and on setting menu-> print speed] [2: Difficult mod. Max.print Speed must be up to 450 (20. Parametic and on setting menu-> print speed]	0-2
P13	[Factory Setting] (0-30)	0-30

P14	Ribbon break active or inactive (0-1) [0: Ribbon break passive] [1: Ribbon break active]	0-1
P15	Print head printing stop position. (0-400) [If the last out of the template does not transfered, value is increased.]	0-500
P16	Pre-heating system active or inactive selection (0-1) [0: Pre-heating system active] [1: Pre-heating system inactive]	0-1
P17	Print head pressed distance before printing (0-50)	0-50
P18	Printer fuse error warning active or inactive setting (0-3) [0: Main board fuse warning ACTIVE, Motor driver fuse warning ACTIVE] [1: Main board fuse warning INACTIVE, Motor driver fuse warning ACTIVE] [2: Main board fuse warning ACTIVE, Motor driver fuse warning INACTIVE] [3: Main board fuse warning INACTIVE, Motor driver fuse warning INACTIVE]	0-3
P19	Fault output mode (0-1) [0: Machine gives alarms ERROR times and inactive times] [1: Machine gives alarm ERROR times]	0-1
P20	Max Print speed (150-700)	150-700

Table-6) 32x40/50I and 53x40/50 I Parameters Explanation

9.Warnings

Savema printer have two types of communication. These are Ethernet(TCP/IP) and RS-232. RS-232 allow max. 115200 bps for communication but Ethernet is faster than RS-232. So, long commands with parameters, can transfer lately in RS-232.

Some commands are need time in printer side.(Load template, Chnage system paramaters …etc). So, connected device must wait until finish processing of this command.

Modification commands allows to change selected object while printer printing.So, large amount of objects can update lately. For prevent this issue, variable object size adjust according to print per minute. Each variable things needs time to update. So, all variable objects must be measured according top rint per minute
Variable objects are;

- Date(Change according to sytem date)
- Time(Change accoding to system time)
- Counter (Changes according to each print)
- Shift Code(Changes according to shifting time)
- Variable Text (Change according to external data)
- Variable Barcode (Change according to each print or external data)
- Variable 2D Barcode (Change according to each print or external data)

Because of the character limitations of the label and command format, following characters must not be used. Use the equivalent form of the character instead.

Character	Equivalent	Converted Form	Example Form
“	"	abc"def	abc”def
‘	'	abc'def	abc’def
<	<	abc<def	abc<def
>	>	abc>def	abc>def
&	&	abc&def	abc&def

Table-7) Character limitations

9.1 Command Limitations

Some Commands are limited to use according to below table.

Command	Explanation	Supported By (Printer Types)
SPCSPS	Set Print Speed	All Intermittent and All Traverse Printers 32x40I, 32x50I, 32x70I, 53x40I, 53x50I, 53x70I, 107x75I, 107x125I, TR32, TR53, TR107
SPCGPS	Get Print Speed	All Intermittent and All Traverse Printers 32x40I, 32x50I, 32x70I, 53x40I, 53x50I, 53x70I, 107x75I, 107x125I, TR32, TR53, TR107
SPCSIC	Set Internal Contact Mode	All Continuous Printers 32C, 32C with Cassette, 32X250C, 53C, 53x250C, 107C
SPCGIC	Get Internal Contact Mode	All Continuous Printers 32C, 32C with Cassette, 32X250C, 53C, 53x250C, 107C
SPCSTC	Set Trigger Contact Mode	All Continuous Printers 32C, 32C with Cassette, 32X250C, 53C, 53x250C, 107C
SPCGTC	Get Trigger Contact Mode	All Continuous Printers 32C, 32C with Cassette, 32X250C, 53C, 53x250C, 107C
SPPOTP	One Test Print	All Intermittent and All Traverse Printers 32x40I, 32x50I, 32x70I, 53x40I, 53x50I, 53x70I, 107x75I, 107x125I, TR32, TR53, TR107
SPGGRR	Get Remaining Ribbon	All with casette modells 32x70I, 32C wth Casette, 53C, 53X250C, 53x70I, 53x125I, 107x75I, 107x125I, 107C
SPTSPS	Set Pack Size	All Traverse Printers TR32, TR53, TR107
SPTGPS	Get Pack Size	All Traverse Printers TR32, TR53, TR107
SPTSPC	Set Print Count	All Traverse Printers TR32, TR53, TR107
SPTGPC	Get Print Count	All Traverse Printers TR32, TR53, TR107
SPTSPP	Set Print Position	All Traverse Printers TR32, TR53, TR107
SPTGPP	Get Print position	All Traverse Printers TR32, TR53, TR107
SPTSPD	Set Pack Distance From Beginning	All Traverse Printers TR32, TR53, TR107
SPTGPD	Get Pack Distance From Beginning	All Traverse Printers TR32, TR53, TR107
SPTSPA	Set Printing Area	All Traverse Printers TR32, TR53, TR107

SPTGPA	Get Printing Area	All Traverse Printers TR32, TR53, TR107
SPTSTP	Set All Traverse Parameters	All Traverse Printers TR32, TR53, TR107
SPTGTP	Get All Traverse Parameters	All Traverse Printers TR32, TR53, TR107

Table-8) Command Limitations

9.2 Command Operating Conditions

COMMAND LIST		
Command	**Explanation**	**Operating Condition**
CONFIGURATION COMMANDS		
SPCSDT	Set System Date&Time and Time Offset	Stop Position
SPCGDT	Get System Date&Time and Time Offset	Both (Print and Stop position)
SPCSNC	Set Network Configuration	Stop Position
SPCGNC	Get Network Configuration	Both
SPCSSC	Set RS-232 Configuration	Stop Position
SPCGSC	Get RS-232 Configuration	Both
SPCSPS	Set Print Speed	Stop Position
SPCGPS	Get Print Speed	Both
SPCSPD	Set Print Delay value	Both
SPCGPD	Get Print Delay value	Both
SPCSDV	Set Darkness(Contrast) Value	Both
SPCGDV	Get Darkness(Contrast) Value	Both
SPCSPR	Set Print Rotation	Stop Position
SPCGPR	Get Print Rotation	Both
SPCSHP	Set Horizontal Position	Both
SPCGHP	Get Horizontal Position	Both
SPCSMO	Set Mirroring Option	Stop Position
SPCGMO	Get Mirroring Option	Both
SPCSRS	Set RibbonSave Mode	Stop Position
SPCGRS	Get RibbonSave Mode	Both
SPCSIC	Set Internal Contact Mode	Stop Position
SPCGIC	Get Internal Contact Mode	Both
SPCSTC	Set Trigger Contact Mode	Stop Position
SPCGTC	Get Trigger Contact Mode	Both
SPCSAS	Set All Settings	Both
SPCGAS	Get All Settings	Both
SPCSSP	Set System Parameter	Stop Position
SPCGSP	Get System Parameter	Both
SPCSPA	Set All System Parameters	Stop Position
SPCGPA	Get All System Parameters	Both
SPCSSL	Set System Language	Both

SPCGSL	Get System Language	Both
SPCSAP	Set Administrator Password	Both
SPCGAP	Get Administrator Password	Both
SPCSFS	Return to Factory Settings	Both
SPCSPM	Set Print Request Message	Both
SPCGPM	Get Print Request Message	Both
LABEL DESIGNING COMMANDS		
SPLTDS	Create Template Datas and Template Structure	Stop Position
SPLLTF	Load Template from Printer	Stop Position
SPLGAT	Get Active Template	Both
SPLGST	Get Stored Templates	Both
SPLCDF	Create Data File	Both
SPLGSD	Get Stored Data Files	Both
SPLDTF	Delete Template	Both
SPLDTA	Delete All Template	Both
SPLDDF	Delete Data File	Both
SPLDDA	Delete All Data File	Both
SPLCDB	Clear Data Buffer	Both
SPLGFN	Get Field Names	Both
MODIFICATION COMMANDS		
SPMCTV	Changing Text Value	Both (Look at the Note at Modification Commands)
SPMCBV	Changing Barcode Value	Both (Look at the Note at Modification Commands)
SPMC2D	Changing 2D Barcode Value	Both (Look at the Note at Modification Commands)
SPMCCV	Changing Counter Value	Both (Look at the Note at Modification Commands)
SPMCSV	Changing Selected Values	Both (Look at the Note at Modification Commands)
PRINT COMMANDS		
SPPSAP	Start Automatically Print	Stop Position
SPPSLQ	Set Print Count for Limited print	Both
SPPGLQ	Get Print Count for Limited print	Both
SPPSTP	Stop Print	Print Position
SPPOTP	One Test Print	Print Position(without Cassette- Intermittent models) Stop Position(with Cassette- Intermittent models)
SPPSTA	Status of Printer	Both
GENERAL COMMANDS		
SPGSUM	Send User Message to Printer	Both
SPGGTP	Get Total Print Count	Both
SPGGCP	Get Current Print Count	Both
SPGGFW	Get Firmware Version	Both
SPGGRR	Get Remaining Ribbon	Both
SPGGSN	Get Serial Number of Printer	Both
TRAVERSE COMMANDS		
SPTSPS	Set Pack Size	Stop Position
SPTGPS	Get Pack Size	Both
SPTSPC	Set Print Count	Stop Position
SPTGPC	Get Print Count	Both
SPTSPP	Set Print Position	Stop Position

SPTGPP	Get Print position	Both
SPTSPD	Set Pack Distance From Beginning	Stop Position
SPTGPD	Get Pack Distance From Beginning	Both
SPTSPA	Set Printing Area	Stop Position
SPTGPA	Get Printing Area	Both
SPTSTP	Set All Traverse Parameters	Stop Position
SPTGTP	Get All Traverse Parameters	Both